

ATARI
ST COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
Juni '87
DM 6,-
Ös 53,-/Sfr. 6,-
6
GUTE ALTE ZEIT ADÉ
Textverarbeitung mit dem ST
FLOPPY KURS
Die Geheimnisse der runden Scheibe
KLICK
Komfort in Dialogboxen
BETRIEBSSYSTEM
Änderung im TOS
BASIC
Für Einsteiger und Profis
MESSEBERICHT
ATARI-SHOW IN LONDON

Am Anfang war immer der Mensch oder ein Plädoyer für die Menschheit.

Werkzeuge haben von Anfang an eine große Rolle in der Entwicklung der Menschheit gespielt. Werkzeuge vereinfachen die Aufgaben, die der Mensch sich gestellt hat.

Die Nutzung dieser Werkzeuge war und ist nicht immer eindeutig gewesen. So wurden Werkzeuge zum Beispiel nicht nur zur Erleichterung der täglich anfallenden Arbeiten benutzt, sondern auch als kriegerisches Instrument gegen andere Stämme. Der Mensch bestimmt also den Einsatzbereich seines Werkzeuges.

Die Werkzeuge haben sich je nach Aufgabenstellung bzw. nach Schwierigkeit der zu lösenden Probleme im Laufe der Zeit geändert. Diese Änderung liegt aber nicht an der Natur des Instrumentes, sondern eher an der intellektuellen Entwicklung des Menschen, sowie an den wachsenden Anforderungen, welche seine Umwelt an ihn stellt. In der heutigen Zeit sind es die zu lösenden Probleme, sowie ihre Komplexität, die die Entstehung und den Einsatz von immer moderneren und effektiveren Werkzeugen bedingt haben. Ohne die Problematik, die von uns Menschen gestellt wird, ist die Entstehung eines neuen Werkzeuges undenkbar.

Im Zeitalter der Computer hat sich an dieser Überlegung auch nicht viel geändert. Die Problematik war vorhanden. Folglich wurde dazu ein Werkzeug, in diesem Fall der Computer, entwickelt, das rationell und effektiv eine Lösung finden kann. Das Einsatzgebiet des Computers war auch nicht von vornherein definiert und erhält schon heute in viele, damals nie für möglich gehaltene Bereiche Einzug. Daß auch die Rechner sich weiterentwickelt haben, liegt nicht an einer Eigendynamik dieser Geräte, sondern an der Fähigkeit des Menschen



seine eigenen Produkte noch verbessern und verfeinern zu können.

Es hat sich historisch erwiesen, daß der Mensch eine fast mystische Beziehung zu solchen Instrumenten hat. So war es bei der ersten Steinaxt und so bleibt es heute noch bei der Verwendung von Computern. Die Einführung der alten Dampfmaschine löste damals große Kontroversen aus zwischen denjenigen, die diese Maschine für unbrauchbar erklärten und den Anderen, die dachten, daß dadurch der Einsatz des Menschen überflüssig geworden sei. Die massive Einführung von Computern in Produktions- und Dienstbereichen macht die Kontroverse von gestern wieder aktuell. Der Einsatz von Computern in unserer modernen Welt ist durch den effektiven Aufwand unserer Entwicklung, sowie die Menge aller Aufgaben, die der Mensch noch zu lösen hat, zu rechtfertigen. Was man dabei allerdings nicht vergessen darf, ist, daß diese Maschine dazu geschaffen wurde, dem Menschen zu dienen und nicht diesen zu ersetzen. Der Mensch selbst, seine Kreativität und seine Entscheidungsfähigkeit, muß unersetzbar bleiben. Dies muß der Menschheit bewußt sein, wenn sie die Macht über den Computer nicht verlieren will.

Damals als die Dampfmaschine eingeführt wurde, wurde der Mensch nicht ersetzt, heute bei der Nutzung von Computern sollte das auch nicht der Fall sein.

In dieser Kette zwischen Problemstellung und Lösung war, ist und wird der Mensch immer am Anfang stehen.

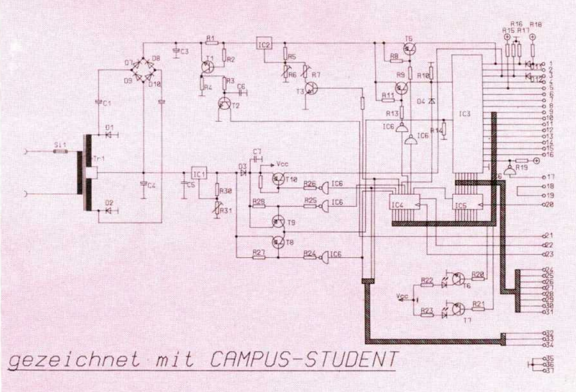
Marcelo Merino

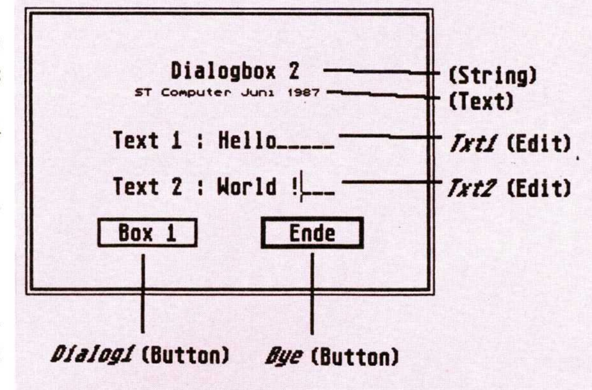
Allgemeines

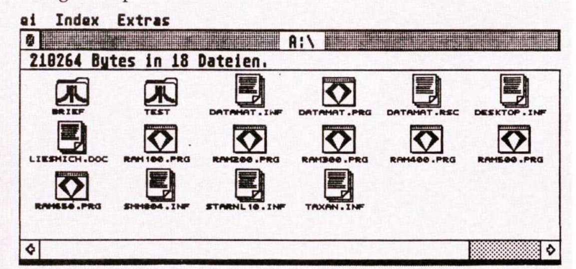
Software



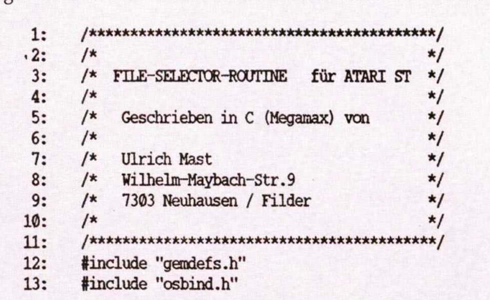
Listing







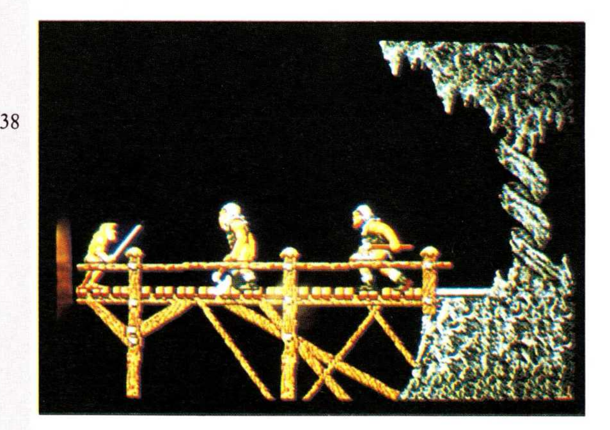
Pfeifenkopfdemo



```
 1:  /*******************************************/
 2:  /*                                         */
 3:  /*  FILE-SELECTOR-ROUTINE   für ATARI ST   */
 4:  /*                                         */
 5:  /*   Geschrieben in C (Megamax) von        */
 6:  /*                                         */
 7:  /*   Ulrich Mast                           */
 8:  /*   Wilhelm-Maybach-Str.9                 */
 9:  /*   7303 Neuhausen / Filder               */
10:  /*                                         */
11:  /*******************************************/
12:  #include "gemdefs.h"
13:  #include "osbind.h"
```

Grundlagen

Kurse



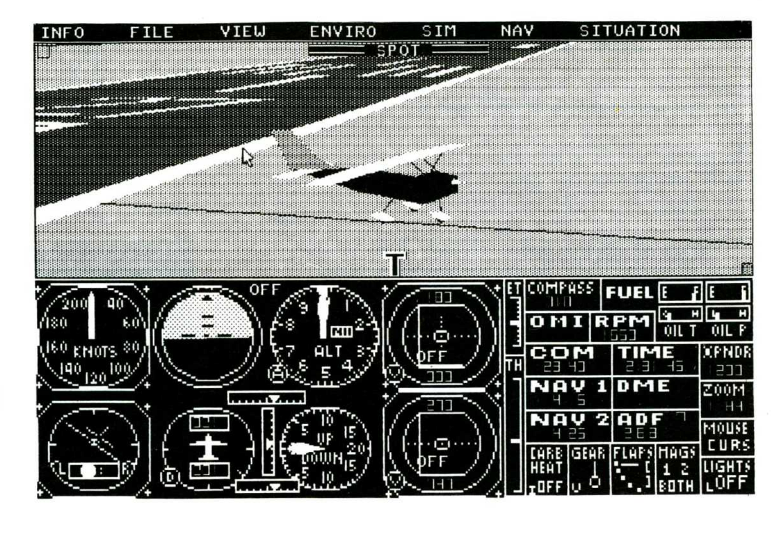
Aktuelles



Kommen die neuen Mega Modelle?

Jetzt muß es sich zeigen, ob die Mega STs, wie angekündigt, Mitte Juni auf dem deutschen Markt erscheinen. Eines steht jedoch fest, der kleinste der drei neuen Mega REchner, der Mega ST 1, wird vorläufig nicht kommen. Der Mega ST 2 (2 MByte RAM-Speicher) und der Mega ST 4 (4 MB RAM) sollen noch im Juni kommen.

Atari Corp. (Deutschland) GmbH
Frankfurter STr. 89-91
6069 Raunheim



Flight Simulator II in monochrom

Der bekannt eFlugsimulator Flight Simulator II von Sublogic ist nun auch in einer Schwarzweiß-Version erhältlich. Das Programm kostet inklusiv deutscher Anleitung DM 179,–.

Softline
Schwarzwaldstraße 8a
7602 Oberkirch

Elektronisches Farbfilter für Digitizer

Das elektronische Farbfilter **PAL-RGB1** ist speziell für Video-Digitizer entwickelt worden, bei denen die Farbdigitalisierung mit Hilfe optischer Farbfilter erfolgt. Moderne Elektronik sorgt für eine einwandfreie Trennung der drei Grundfarben (Rot, Grün, Blau) und für eine Farbsättigung, die mit optischen Filtern nicht zu erreichen ist. Da im Ausgangssignal kein Farbhilfsträger (Burst) mehr enthalten ist, können auch Digitizer angeschlossen werden, die den Anschluß einer Schwarzweiß-Kamera vorschreiben.

Der RGB-Farbfilter wird einfach in die Videoleitung zwischen Farbkamera und Digitizer geschaltet. Die Anwahl der drei Grundfarben kann



manuell oder automatisch durch den Computer erfolgen. Die Intensität der einzelnen Farben kann mit Hilfe eines Reglers eingestellt werden.

Zum Vergleich der Qualität ist hier das Originalbild unseres Digitizer-Vergleichs in Heft 4/87 (S. 14) mit dem PAL-RGB1 Filter und dem Realtizer von Print Technik digitalisiert worden.

Creative Video
Postfach 15 01
8520 Erlangen
Tel. 0 91 95 / 27 28

Noch ein Digitizer

Video-1000 ST heißt ein neuer Monochrom- und Farb-Digitizer. Bei Anschluß des Monochrom-Monitors (SM 124) können Bilder mit 2, 3 und 5 Graustufen digitalisiert werden. Im Farbmodus können Bilder mit 4, 5, 8 oder 16 Farben wiedergegeben werden. In der Schwarzweiß-Darstellung wird eine Auflösung von 640 x 400 Pixel erreicht. Das Gerät ist in einem Kunststoffgehäuse mit den Abmessungen 12 x 6 x 2 cm untergebracht und wird an den ROM-Port des ATARI ST angeschlossen. Zum Lieferumfang gehört das Interface samt Software sowie ein Konvertierungsprogramm auf Degas- und Neochromformat. Der Video-1000 ST kostet DM 295,–.

Ing. Büro Manfred Fricke
Neue Straße 13
1000 Berlin 37
Tel. 0 30 / 901 56 52

Nachlese zu unserem Bericht Digitizer im Vergleich aus Heft Nr. 4/87

Nach Rücksprache mit der Firma Print Technik, haben wir einen falschen Begriff für unsere Zeitmessungen verwendet. Während wir in unserem Bericht als Digitalisierungszeit etwa vier Sekunden für 16 Graustufen beim Realtizer angegeben hatten, so benötigt dieser, laut Herstellerangaben, bei 16 Graustufen nur 0,64 Sekunden zur **Digitalisierung**. Dieser Zeitunterschied ist folgendermaßen zu erklären: Wir haben die Zeit angegeben, die zum vollständigen Bildaufbau des digitalisierten Bildes benötigt wird. Bei allen drei getesteten Geräten haben wir selbstverständlich nach der gleichen Methode gemessen. Die Firma Print Technik gibt als Digitalisierungszeit die reine Aufnahmezeit für das Videobild an (Umwandlungszeit des A/D-Wandlers), ohne den Bildschirmaufbau zu berücksichtigen. Diese Zeitangabe ist durchaus sinnvoll, läßt sich jedoch nicht so leicht nachmessen, so daß wir sie nicht erwähnt hatten. Wir hoffen, daß dieses Mißverständnis nun behoben ist.

Kodierungsprogramm für wichtige Daten

Um dem allgemeinen Wunsch nach Datensicherheit gerecht zu werden, in einer Zeit, in der selbst schon neuentwickelte Software und Source-Code unter der Hand veräußert werden, ist dieses neue Softwarepaket entwickelt worden. Außer der Kommunikation mit anderen Terminals können nun auch interne Daten für jeden Unbefugten sicher abgelegt werden. Selbst das Versenden wichtiger Daten per Post oder Boten ist für jede Art der Anwendung per Diskette möglich. Die Software, die als Accessory und gewöhnliches Programm geliefert wird, arbeitet nach dem ISO und ANSI Standard. Das Programm kostet DM 198,–. Über das Thema Datensicherung werden wir in einer der nächsten Ausgaben noch ausführlich berichten.

Krypto-Soft GmbH
Weizenfeld 36
3060 Berg. Gladbach 2
Tel. 0 22 02 / 3 06 02

Ein neues CAD Programm

MAG-Multi-CAD heißt ein CAD-System, das unter anderem folgende Leistungsmerkmale aufweist:
- Vollbildfadenkreuz und punktweise beliebiger Rastereinstellung
- Software voll in GEM eingebunden
- Zeichengröße bis DIN A4
- Umfangreiche Bildteiloperationen, wie Drehen, Verschieben, Kopieren ect. sowie eine Lupe sind vorhanden
- Indexeinstellungen, wie Füllmuster, Linien, Text, Raster, Kopiermodus und Radierer
- Einfache Druckeranpassung an fast alle grafikfähigen Drucker (auch 24 Nadler) möglich.

MAG-Multi-CAD eignet sich zum Erstellen technischer Zeichnungen, Konstruktionen, Platinenlayouts ect. und kostet DM 149,–.

MAG-Software
Matthias Gärtner
Schwarzwaldring 49
7505 Ettingen 4
Tel. 0 72 43 / 2 84 06

ATARI ST und SHARP Pocketcomputer

Die Firma Yellow Computing bietet ein Interface samt Software an, mit dem es möglich ist folgende Pocketcomputer der Firma SHARP mit dem ATARI ST zu verbinden und somit einen Datentransfer zwischen diesen Rechnern herzustellen:
- PC 1260/61/62,
- PC 1401/02/03/21/30/50,
- PC 1350/60

Die in GEM eingebundene Software sorgt für ein bequemes Editieren Ihrer SHARP-Programme auf dem ST. Ein bidirektionales Übertragen von BASIC-Programmen, Speicherbereichen und Variablen ist möglich. Listings können über den ATARI ST gedruckt werden. Das Interface kostet inklusiv der Software DM 99,–.

Yellow Computing
Wolfram Herzog
Im Weingarten 21
7101 Hardthausen-Lampoldshausen

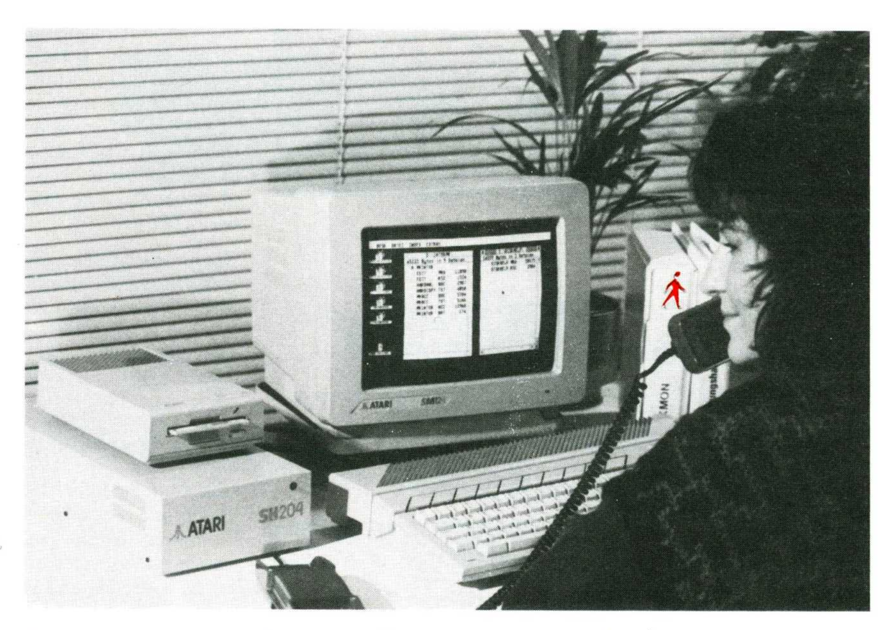
PD-News

Auch im Bereich der Public-Domain Software möchten wir Ihnen wieder Programme vorstellen. Aus Platzgründen können wir leider nur einige der vielen erwähnen.

Thomas Proffen, mit Künstlernamen 'Wurmsoftware', schrieb **SYNTHI**. Dieses Programm ist ausgelegt zum Erzeugen von Soundsequenzen, wobei sämtliche Einstellungen auf einfache Art und Weise erfolgen. Aus der Hardcopy sind einige Möglichkeiten erkennbar, doch erst nach mehrmaligem Probieren und dem Anhören der Demos, kann man sich ein Bild der Einsatzmöglichkeiten von 'Synthi' machen. Wir empfehlen es zur Unterlegung eigener Programme mit Geräuschen oder einfacher Musik, da es nach Fertigstellung einer Soundsequenz, diese so als File ablegt, daß sie ohne Probleme vom eigenen Programm geladen und gestartet werden kann. Dazu sind Demos vorhanden, die das Einbinden des Sounds beschreiben. Neben den Demos ist auch der komplette Source-Code in Pascal enthalten (PD 60).

Haben Sie schon einmal Disketten mit einer Hardcopy beruckt? Wie denn ist doch viel zu groß! Falsch gedacht, mit der **Hardcopyroutine** von Michael Beuten ist dies kein Problem. Die Grö-

Wir machen Spitzentechnologie anwenderfreundlich



Programme, die auch Frauen bedienen können.

INTERPRINT DM 49,—

Die flexible Druckeranpassung für Ihren ST. Nicht nur Text, auch Hardcopies können angepaßt werden. Mit Druckerspooler auch für Hardcopies. Auch mit integrierter Ramdisk: DM 99,—.

DISK HELP DM 79,—

Mit DISK HELP können Sie sowohl defekte Daten retten als auch Lesefehler beheben, solange kein mechanischer Diskettenschaden vorliegt.

AS SOUND SAMPLER DM 79,—

Die komfortable Software ermöglicht nicht nur das Aufnehmen und Abspielen von Sounds, sondern auch Effekte wie Stottern, Hall usw. Fertige Sounds können in Programme eingebunden werden. 58 KHz, 8 Bit AD/DA Wandler. Mit Bauanleitung. 2 Demodisketten: DM 10,—. Fertiggerät: DM 129,—.

G RAMDISK II DM 49,—

Bis 3,9 MB resetresident auch nach Systemabsturz. Mit integriertem Druckerspooler. Voll abschaltbar.

G DISKMON II DM 89,—

Sehr komfortabler GEM Diskettenmonitor mit vielen Sonderfunktionen für Direktzugriff. Ideale Ergänzung zu DISK HELP.

HARDDISK HELP & EXTENSION

Das Backupsystem der Spitzenklasse. Sichert alle gewünschten Daten von der Harddisk in kürzester Zeit auf Disketten. Mit Sonderkriterien.

DM 129,—

G DATA



Siemensstr. 16 · 4630 Bochum 1

TELEFONISCHE BESTELLUNGEN: 023 25 / 6 00 98

Verkauf: Industriestraße 40
4630 Bochum 7

Schweiz: Senn Computer AG
Langstraße 21 · CH-8021 Zürich

Bestell-Coupon

Bitte senden Sie mir:

zzgl. DM 5,— Versandkosten

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name

Straße

Ort

ße liegt bei (68 mm ★ 48 mm) und liefert dabei eine wirklich gute Qualität. Lauffähig ist diese Routine sowohl auf 9- als auch auf 24 Nadeldruckern, wobei sie bei 24-Nadlern vertikal ein wenig größer wird. In beiden Fällen ist die Größe ideal für Etikettenaufdruck (PD61).

DATOBERT hat keineswegs etwas mit Walt Disney zu tun, obwohl es der fast gleichnamige, hyperreiche Entenhäusler gut zur Übersicht seiner Geldspeicherfüllstände gebrauchen könnte. Das Programm von J. Fiedler ist zur grafischen Darstellung von Zahlen geschrieben; dabei bietet es die Darstellungsarten: Kuchen-, Linien-, Balken-, Säulen- und Flächengrafik. Die beiden letzteren erlauben die dreidimensionale Darstellung von mehreren Werteserien, die zu einem sehr anschaulichen Überblick führt. Der Source in GFA-Basic ist beigefügt (D61).

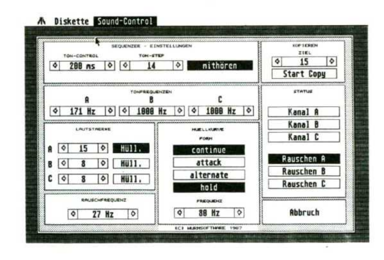
Im Bereich der Spieleszene freuen wir uns, Ihnen einige neue Werke zu präsentieren. **'Das dreimal verflixte Quadrat'** von Dietrich Raisin (Pancreation) ist eine Umsetzung des bekannten Zahlenpuzzlespiels, bei dem man auf 16 Felder 15 Steinchen in die richtige Reihenfolge bringen muß. Wie Sie sehen können, wurden diese Steinchen durch ein Bild ersetzt. Dadurch gewinnt dieses Spiel an Reiz aber auch an Schwierigkeit (PD 66).

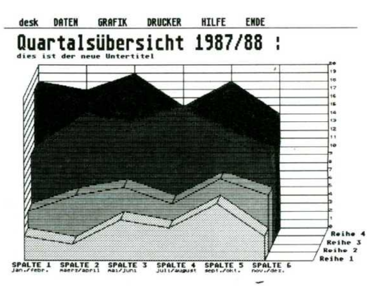
Die Freunde des Würfelspiels können sich über **'FUFFY'** von Rudolf Brueck und Jürgen Neervoort freuen.

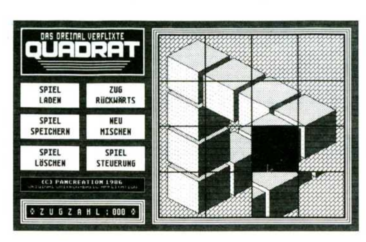
Dieses Spiel ist inhaltlich vergleichbar dem bekannten KNIFFEL aus der 'Nicht-Computer Szene'. Im letzten Monat stellten wir bereits ein ähnliches Würfelspiel vor, dieses wurde aber um einige Punkte erweitert. So lassen sich nach einigen Spielen anschauliche Statistiken über den Würfel- und Spielverlauf anzeigen. Auch die übrige Gestaltung ist sehr schön programmiert und wird zum Spielen empfohlen (PD 63).

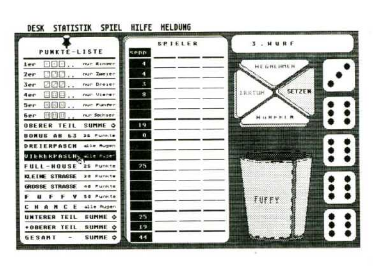
Über das umfangreichste Programm **'MAZIACS'** werden Sie in der Sparte RELAX informiert.

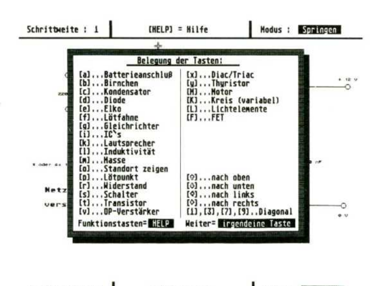
Das Zeichnen elektronischer Schaltpläne ist die Aufgabe des Programmes **'E-PLAN'**. Per Tastendruck erscheinen die angewählten Bauteile auf dem Bildschirm und zwar in der jeweiligen Richtung in der man sich gerade auf dem Zeichenbrett bewegt. Welche

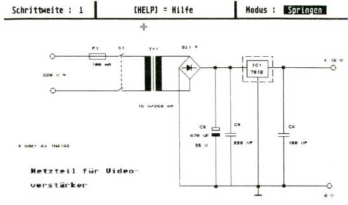












Teile vorgesehen sind, ist einem Menüpunkt des Programmes zu entnehmen. Somit läßt sich eine kleine Schaltung sehr schnell auf den Bildschirm zaubern, wofür auf Blatt und Papier schon einige Zeit vergeht. Die Beschriftung der fertigen Zeichnung ermöglicht das Programm von Ralf Neußinger ebenfalls, sodaß es für alle Hobbybastler eine nützliche Hilfe sein wird (PD 64).

Doch nicht nur über neue Programme ist zu berichten, auch über ältere, die verbessert wurden, sollten einige Worte fallen. Da wäre das Programm **Finanzmathe** (PD 32). Dieses wurde stark erweitert und hat vieles zu bieten. Dazu zählen:

vor- und nachschüssige Zins- und Rentenrechnung, Effektiv- und Nominalzins, Anuitäten und Ratentilgung, Diskontierung und Konvertierung. Das Programm wurde gegenüber seinem Vorgänger schneller und ist aufgrund der Programmierung in OMIKRON-Basic, mit höherer Genauigkeit ausgestattet.

Sämtliche Public-Domain Disketten sind direkt über den PD Service der Redaktion zu beziehen. Näheres entnehmen Sie bitte den gesonderten Public-Domain Seiten.

ST-Computer Redaktion
Schwalbacherstr. 64
6236 Eschborn
Tel. 0 61 96 / 48 21 58



Die ST Computer und die Firma NEC Deutschland GmbH rufen gemeinsam alle Programmierer zu einem Wettbewerb auf. Dabei sind drei 24-Nadeldrucker zu gewinnen.

Folgendes Programm für den ST ist zu erstellen:

Eine Hardcopy-Routine, die einen luxuriösen Ausdruck auf den NEC-Pinwriter-Druckern ermöglicht. Das Programm soll mindestens folgende Features enthalten:

- Aufruf über Alternate/Help.
- Danach soll über eine Auswahlbox die Möglichkeit gegeben sein, Optionen einzustellen, direkt zu drucken oder den Vorgang abzubrechen.
- Als Optionen sind sowohl Farb- als auch S/W-Hardcopy gefordert (180 bzw. 360 Dots). Ferner sollte ein Download von Zeichensätzen in den Drucker möglich sein.

Neben diesen Bedingungen sind natürlich alle anderen Ideen für die kreative Gestaltung eines solchen Programmes zugelassen. Die Wahl der Programmiersprache bleibt Ihnen freigestellt. Wir benötigen das ausgedruckte Listing und das Programm auf Diskette. Wenn Sie einen ausreichend frankierten Rückumschlag beilegen, wird die Diskette nach der Siegerentscheidung zurückgesandt. Das Programm muß natürlich selbst erstellt und frei von Rechten Dritter sein. Die zehn besten Programme werden auf einer Sonderdiskette im Public-Domain-Service dieser Zeitschrift veröffentlicht.

Die Auslosung erfolgt unter Ausschluß des Rechtsweges.

Die Preise:

Platz 1: Ein NEC P5 XL Farb-S/W-Drucker im Wert von DM 3600,–

Platz 2: Ein NEC CP7 Farb-Drucker im Wert von DM 3060,–

Platz 3: Ein NEC CP6 Farb-Drucker im Wert von DM 2400,–

Platz 4: Ein Einzelblattschachteinzug für P6 oder P7 im Wert von DM 860,– bzw. DM 1070,–

Plätze 5 und 6: Ein bidirektionaler Traktor für P6 und/oder P7 im Wert von DM 348,– bzw. DM 448,–.

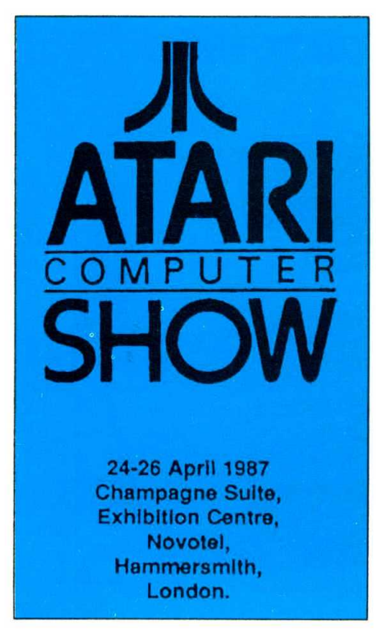
Platz 7: Ein ST Computer Abonnement für die Zeitdauer eines Jahres.

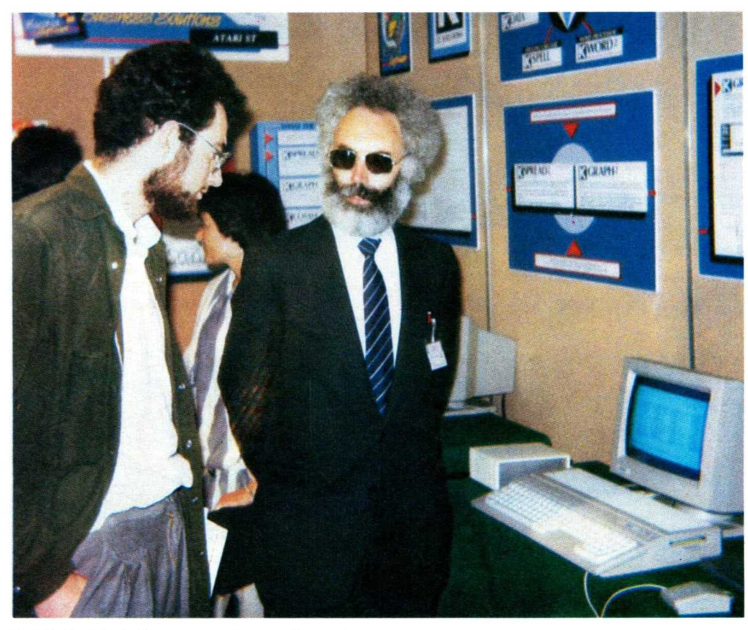
Plätze 8 — 10: Jeweils zehn Disketten, bespielt mit Public-Domain-Software nach Wahl.

Schicken Sie bitte Ihr Programm an folgende Adresse:

ST Computer
Stichwort: Luxus-Hardcopy
Schwalbacherstr. 64
6236 Eschborn

Einsendeschluß ist der 31. August 1987




Bild 2: Gedränge am Kuma-Stand

Für den britischen ST-Besitzer gibt es zwei wichtige Termine im Jahr:

- **die im September stattfindende PCW Show im Londoner 'Olympia' (eine kleine englische CeBIT) und**
- **die ATARI Computer SHOW, die mit mehr als 2000 qm Ausstellungsfläche und – diesmal – 78 anwesenden Soft- und Hardwareproduzenten, Atari-Computermagazinen und diversen Userclubs eine der größten Atarimessen der Welt ist.**



Bild 1: Der Mega ST

Dem deutschen ST-Freak sind – so kurz nach der CeBIT – etliche der Neuerscheinungen schon vertraut, hat er doch bereits in Hannover Vorabversionen davon gesehen oder getestet (vgl. ST 5/87)

Der neue Mega ST

Im Zentrum des Publikumsinteresses stand natürlich der neue Mega St.

Mit einem RAM Memory bis zu 4 Mbytes, einem eingebauten 720K Floppy Disk Drive und der abgesetzten Eingabeeinheit ist er in Leistung und Aussehen ein Leckerbissen. Seine Kompatibilität mit dem 'normalen' ST und das spezielle, auf einfachen Anschluß von Erweiterungen ausgelegte Design, gibt Atari als selbstverständlich an.

Lattice C Compiler V3.04

Brandneu ist die Version 3.04 des Lattice C Compilers der Firma Metacomco. Neben 'Resorce Construction Editor', 'Symbolic Debugger and Dissasembler' werden über 100 neue Libraries angeboten. Metacomco hat uns übrigens fünf Originale zum Verschenken an die ST-Leser mitgegeben. Mehr darüber und über die Möglichkeiten des Programmierens in C ist im nächsten Heft zu lesen.

Desktop Publishing

Wie in Hannover war auch in London der Fleet Street Publisher von Mirrorsoft Trumpf (vgl. ST 5/87). Leider gibt es noch keine deutsche Ausgabe. Eine Mischung aus Desktop Publishing und Wordprocessor soll 'Calligrapher' von Computer Concepts werden. Es ist wie 'Fleet Street Publisher' in GEM eingebunden. Erscheinungsdatum: August.

KUMA

Mit seiner ganzen Palette an Text- und



Bild 3: Apiel: Shuffleboard

Datenverarbeitungsprogrammen war Kuma Software vertreten. Neben den verbesserten Versionen aus der K-Serie, wie Kword2, Kgraph2, Kspread2 und Kdata, wurde diesmal Kmax – ein RISC Processor Development System präsentiert.

Das magische Dreieck

Mit einem großen Stand und entsprechender Angebotspalette war Eidersofts 'Triangle – soft innovation' vertreten. Sie lieferten den Beweis, daß Gutes durchaus nicht billig sein muß. Ein 20MB ST Turbo Drive (Hard Disk) wurde für umgerechnet 1800,– angeboten. Zum halben Preis hingegen war bereits das 'Pro Draw Graphics Tablet' zu erstehen – ein '9 x 6 high resolution graphicstablet', dessen ST Version speziell an den 'art & film director' von Mirrorsoft angepaßt wurde.
Demnächst soll ein 'Sprite-Editor' erscheinen, um den Hobbyprogrammierer beim Animieren und Einbinden von Sprites in selbstgestrickte Programme zu unterstützen. Die in London vorgeführte Beta-Version verspricht gute Grafik bei einfacher Handhabung. Das Erscheinungsdatum ist aber noch nicht bekannt.

Zwei in einem

Bereits erschienen ist hingegen das neue 'Doppelpaket' von Robtek. Es beinhaltet diesmal die Spiele 'Shuffleboard' und 'ST Pool'. 'ST Pool' ist ein relativ kompliziertes Billard-Spiel mit guter Grafik, aber schwieriger Steuerung. 'Shuffleboard' hingegen ist sehr einfach zu spielen. Es stellt eine Art Curling auf einer Holzbahn dar. Die Idee, zwei Spiele zusammen und zum Preis von einem zu verkaufen, ist sicher im Sinne des Konsumenten.

Spiele, Spiele, Spiele...

Computermessen ohne neue Spiele sind nicht mehr denkbar. Immerhin muß der ST-Besitzer jetzt nicht mehr neiderfüllt in Richtung Amiga schielen, wenn er ein Actionspiel mit excellenter Grafik sehen möchte.
Wegweisend ist hier Microdeal mit Airball und Goldrunner. Diese Spiele, von denen auf der CeBIT nur Demoversionen erhältlich waren, sind ab sofort im Handel.

Den Weg, ein Spiel billig, aber dafür sehr oft zu verkaufen, hat Tynesoft jetzt eingeschlagen. Man bietet unter dem Label 'MicroValue' die Spiele 'Plutos' und 'Mousetrap' zum Preis von ungefähr DM 30,– an. Das Äußere ist im Gegensatz zu Produkten anderer Firmen etwas spartanisch, aber wer spielt schon mit der Verpackung? 'Plutos' ist ein Weltraumschießspiel mit guter Grafik; 'Mousetrap' erinnert an die Plattformspiele alten Stils, wobei sich im Gegensatz zum guten alten Commodore 64 zwar nicht die Idee, aber die Graphik gewandelt hat.

Gleich mehrere positive Überraschungen hatte Psygnosis zu bieten. Erstens gibt es endlich eine Joystickversion von Arena. Zweitens gibt es demnächst ein neues Spiel mit absoluter Spitzengrafik – 'Barbarian'. Es ist die Geschichte von Hegor, dem Barbaren, der auszieht, die Welt Durgan von Necron, dem bösen Tyrannen, zu befreien. Als Preis winkt ihm die Königskrone. Unseren Lesern winkt – und das war die dritte positive Überraschung – die Verlosung von zehn Originalen dieses neuen Psygnosis-Hits. Mehr darüber und ausführliche Besprechungen der hier vorgestellten Spiele in unserem nächsten Heft.

P. S. Ein Bonbon am Rande wollen wir natürlich nicht verschweigen: Ein kanadischer Programmierer (Donald ...) hat Microdeal einen Commodore 64-Emulator auf Softwarebasis angeboten. Wir konnten uns davon überzeugen, daß bereits diverse Basicprogramme damit laufen. Weitere Informationen wollte man uns aus verständlichen Gründen im Moment nicht geben.

C. P. Lippert



Bild 4: Gute Grafik bei Barbarian



ATARI®
ST COMPUTER
Die Fachzeitschrift für den ATARI-ST Anwender.
DM 14,–/Ös 112,–/Sfr. 14,–
EXTRA
VIELE
Listings in allen wichtigen Programmiersprachen
ERSTE
Ausführliche Anleitung für das Resource Construction-Set von Digital Research
VIELE
Grundlagenartikel
Das große ST-EXTRA · ab 26. Juni im Handel
Bestell-Coupon Einsenden an: Heim-Verlag, Heidelberger Landstr. 194, 6100 Darmstadt 13
St. ST-EXTRA zum Preis von 14,– DM je Heft
Der Versand erfolgt kostenfrei
Zahlung: nur gegen Übersendung eines Schecks
Name
Straße
Ort

Fernseher die Zweite

Nachdem in der letzten Ausgabe die Bildqualitäten zweier normaler Fernseher am ST beschrieben wurden, wollen wir jetzt noch einmal auf dieses Thema eingehen. Zum einen wird das Testbild näher beschrieben und zum anderen der Anschlußplan des Kabels zwischen Fernseher und ST.



Bild 1: Das Testbild für Monitore

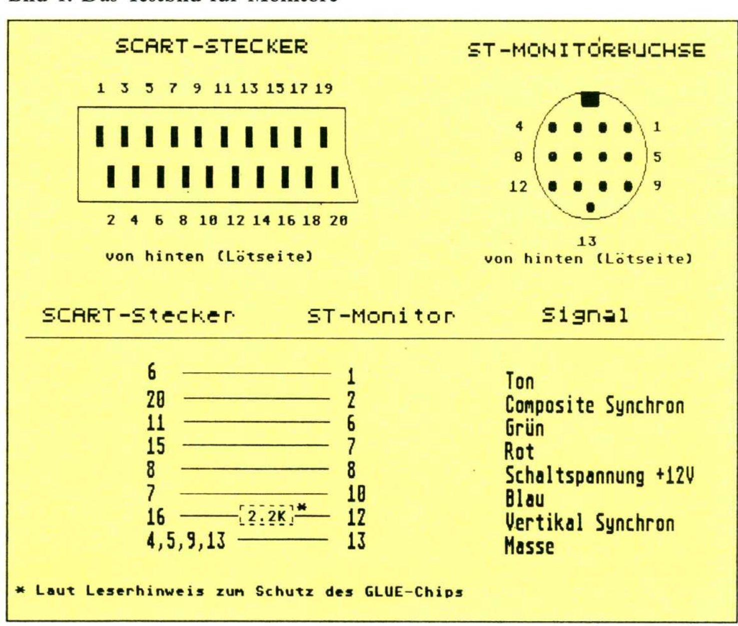


Bild 2: Anschlußbelegung eines SCART-Kabels

Das Testbild

Schaltet man zu gewissen Zeiten den Fernseher ein, so blickt einem nur ein Testbild entgegen, die Sendungen beginnen meist erst später. Doch so uninteressant wie es zunächst scheint, ist ein Testbild nicht. Dem ist auch so beim ST. Will man sein Monitorbild oder die Farben justieren, ist ein genauer Abgleich unerläßlich, das Testbild muß her.

Das oben gezeigt Bild ist auf PD-Diskette 46 und 67 enthalten und kann mit „NEOCHROME" (PD 1) geladen und betrachtet werden.

Diese Kurzanleitung wird Ihnen das Testbild erläutern:

- Die oberen 8 Felder (Grautreppe) helfen beim Einstellen von Helligkeit, Kontrast und Graubalance. Stellen Sie die Regler „Helligkeit" und „Kontrast" so ein, daß das erste Feld weiß und das letzte Feld schwarz erscheint. Außerdem muß das erste vom zweiten bzw. das letzte vom vorletzten Feld zu Unterscheiden sein (Kontrast).

  Sollten die Felder einen Farbstich haben, müssen die Farben rot, grün und blau einzeln eingestellt werden, bis alle 8 Felder neutral wirken.

  Farbfernsehgeräte bieten diese Eingriffsmöglichkeit von außen normalerweise nicht, und der Abgleich der Regler R-Y, B-Y, G-Y bzw. R, B, G auf der Geräterückseite sollte nur von einem versierten Fachmann durchgeführt werden. Es könnte auch sein, daß der Videomodulator ihres Computers diesen Farbstisch erzeugt. Bei einer normalen Fernsehsendung würde sonst der entgegengesetzte Farbfehler auftreten.

  Wenn einige Graustufen bunt bzw. von anderen nicht zu unterscheiden sind, so hat Ihr Videomodulator vermutlich einen Defekt. Sollten alle Felder einen sehr starken Farbstich in der Farbe gelb, dunkelrot oder blau haben, so ist das Verbindungskabel zum Monitor zu überprüfen. Ihrem Monitor „fehlt" eine Farbe.
- Die nächsten 8 Felder zeigen unter anderem die bunten primären und sekundären Farben. Hiermit können Sie die Farbsättigung ihres

Fernsehmonitors einstellen. Geben Sie nur so viel Farbe, daß kein Überstrahlen entsteht.

Ein RGB-Monitor (SCART) braucht diese Einstellung nicht. Die Farbstärke ist von selbst richtig eingestellt.

- Die nächsten Felder geben Aufschluß über die Wiedergabeschärfe des Monitors.

Wenn Sie einen SCART-Anschluß bzw. einen RGB-Eingang benutzen, können immer einzelne Pixel dargestellt werden. Die Wiedergabeschärfe ist stets hervorragend – also größer als 4 MHz –.

Wird der Monitor mit einem Standard-Videosignal versorgt, ist die Schwarzweiß-Wiedergabe weniger scharf. Üblich ist hier eine Schärfe zwischen 3 MHz und 4 MHz. Die Farbwiedergabe ist bei diesem Verfahren stets noch wesentlich weniger scharf. Die Farbbalken verdeutlichen dies.

- Die letzten Felder beweisen noch einmal die Schärfe der getrennten Übertragung von Rot-, Grün- und Blau-Signalen. Mit RGB-Verfahren kann jedes Pixel unterschieden werden, denn hier wird grundsätzlich mehr als die dreifache Bildinformation übertragen. Geht man weit genug vom Bildschirm weg, so sieht man eine Mischfarbe.

Der Video-Monitor mischt diese Farben durch die Unschärfe des Übertragungsverfahrens bereits im Gerät. Das kann auch ein gewisser Vorteil sein, denn es zeigen sich nun mehr als 16 Farben auf einmal. Dreht man den Farbregler auf Null, erhält man auch noch mehr als 8 Grautöne. Allerdings trennt sich in Bezug auf Moiréebildung – das sind schräg durchlaufende streifenartige Muster – bei den Videomodulatoren und den Fernsehgeräten die Spreu vom Weizen.

Einige Modulatoren machen das Bild deshalb „ungenießbar", andere lassen ein Moirée kaum erkennen. Und das ist bei PAL übrigens nicht völlig zu vermeiden.

Viel Erfolg beim Monitorabgleich.

Ralf Bager

Listgeschützte Programme compilieren

GFA Basic-Programme, die mit der PSAVE-Option abgespeichert wurden, sind nicht listbar. Das hat seine Gründe und ist voll akzeptiert, aber manchmal kommt es vor, daß man ein solches Programm compilieren möchte. Der Compiler weigert sich aber, ein 'PSAVE-Programm' zu bearbeiten, ein klitzekleiner Eingriff schafft dabei Abhilfe.

Im Prinzip muß man dazu nur das erste Byte eines listgeschützten Basic-Programmes von 255 auf 00 ändern.

normales Programm 00 02 71 102 ..
geschützte Programme 255 02 71 102 ..

Das zweite Byte stellt die Versionsnummer des GFA-Basics wieder, im oben angegebenen Beispiel handelt es sich um Version 2.0.

Das Programm öffnet nun das Basic-Programm als Datei und setzt den Dateizeiger auf das erste Element. Dieses wird sodann, falls es sich um ein geschütztes Programm handelt, geändert, und die Datei wieder abgespeichert. Das war's. Der Compiler akzeptiert das geänderte File und erzeugt ein ablauffähiges Maschinenprogramm.

(HS)

```
' Modelt das erste Bytes eines listgeschützten GFA-Programms
' so um, daß es vom Compiler akzeptiert wird.
'
' Listen ist immer noch nur teilweise möglich
' HS
'
Fileselect "\*.bas","",Name$
If Exist(Name$)              ! Gibt es das File auf Disk
  Open "U",#1,Name$          ! Schreib/Lese Zugriff
  Seek #1,0                  ! Zeiger auf Anfang
  Nixlist=Inp(#1)            ! erstes Byte lesen
  Version=Inp(#1)            ! zweites Byte lesen
  Print "Dies ist ein GFA-Programm Version ";Version
  '
  If Nixlist=0               ! wenn listen
    Alert 1,"Dieses Programm läßt |sich aber listen",1,"Ach so",Dum
  Else
    If Nixlist=255           ! wenn Listschutz
      Alert 2,"Soll ich den Schutz entfernen",1,"Na Klar|Niemals",Erg
      If Erg=1
        Seek #1,0            ! Zeiger an Anfang
        Out #1,0             ! Listschutz entfernen
      Endif
    Endif
  Endif
  '
  '
  Close #1                   ! Datei schließen und Schluß
Endif
```

Er hat versucht, mit seiner Textverarbeitung zu rechnen...



Demnächst auch für AMIGA und Schneider PC

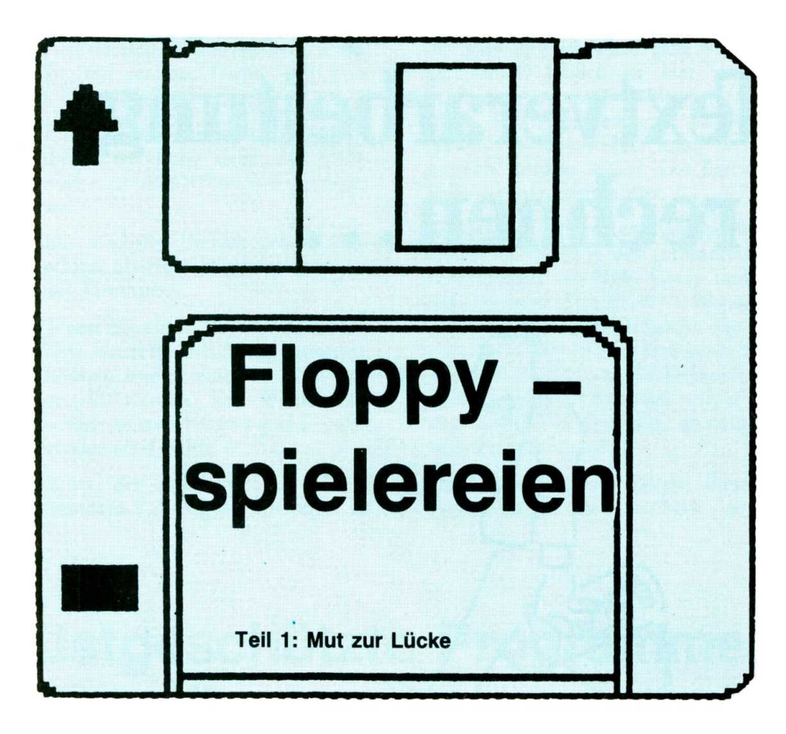
BECKERtext ST

Textverarbeitung, wie sie sein sollte.

Man bräuchte eben eine Textverarbeitung, die nicht nur souverän mit Buchstaben, Wörtern und Textblöcken jonglieren kann, sondern auch mit Zahlen umzugehen weiß. Komfortables Rechnen im Text, nicht nur spalten-, sondern auch zeilenweise, mit bis zu 6 Nachkommastellen bei 10stelliger Genauigkeit – für BECKERtext ST ein Leichtes. Zudem unterstützen Dezimaltabulatoren die Lesbarkeit – bei Tabellen nahezu unverzichtbar.
Möchte man zusätzlich noch Abbildungen verwenden, aktiviert man einfach BTSNAP. So lassen sich mit BECKERtext ST aus fast jedem Programm die gewünschten Bildschirmausschnitte als Datei abspeichern und später an entsprechender Stelle im Text ausdrucken. Bei Farbgrafiken erfolgt automatisch eine farbgerechte Grautonumsetzung. Klingt einfach? Ist es auch – wie alles bei BECKERtext ST. Schließlich ist diese leistungsfähige Textverarbeitung GEM- und mausgesteuert, wobei wichtige Befehle zusätzlich auch über Tastendruck angewählt werden können. Selbst bei der mühseligen, aber notwendigen Korrekturarbeit bietet BECKERtext ST optimale Hilfe. Wahlweise ONLINE während des Schreibens oder nachträglich überprüft das eingebaute Lexikon den Text auf Rechtschreibung. Individuell erweiterbar eignet sich dieses Lexikon auch für Fremdsprachen.
Mehrspaltige Druckausgabe mit bis zu 5 Spalten, automatische Silbentrennung, bis zu 999 Zeichen pro Zeile bei horizontalem Scrolling, schnelle Direktformatierung und Formulare als nicht überschreibbare Eingabemaske sind weitere, hervorstechende Features. Dazu viel Produktpflege mit Liebe zum Detail von 128 zusätzlichen Sonderzeichen über Dezimaltabulatoren bis hin zu ausgeklügelten Druckeranpassungen. Bei alledem ist BECKERtext ST nicht kopiergeschützt und kostet nur DM 199,–.

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
zzgl. DM 5,– Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort
ST 67

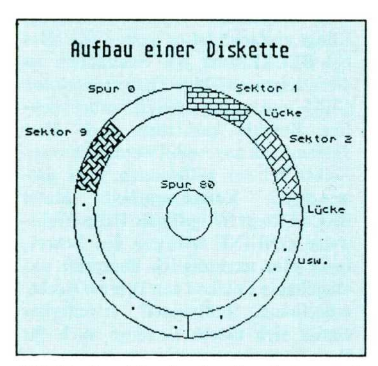


Wie der ST seine Disketten verwaltet, und wie man ihm dabei gehörig auf die Sprünge helfen kann, soll hier am Beispiel eines Programmes gezeigt werden, das Träume erfüllt: Der HYPERFORMATTER bläst Disketten bis auf schwindelerregende 927000 Bytes auf und beschleunigt dazu noch den Zugriff um bis zu 30 %

927000 Bytes?! Wie macht man das? Um das zu erklären, muß ich weiter ausholen, weil mein Programm viele Finessen verwendet, die erst mal erklärt werden wollen. Ich möchte Ihnen dazu in einem Blitzkurs in zwei Folgen Grundlegendes und Raffiniertes über Disketten und Laufwerke erzählen. Sie sollten auch schon mal eine Diskette in der Hand gehabt haben — das sid diese kleinen viereckigen Dinger, die immer dem Bürostuhl unter die Rolle kommen (knirsch) — und Bits von Bytes unterscheiden können, sonst wird dieser Kurs Ihnen wohl nicht so viel bringen. Aber zumindest können Sie die Programmbeispiele abtippen und sich an den Ergebnissen delektieren.

Spuren, Sektoren und Lücken

Beginnen wir zur Abwechslung mal am Anfang: TOS, das Betriebssystem des ST (also nicht dieser Grafikaufsatz namens GEM), hält sich in der Diskettenverwaltung recht sklavisch an Vorgaben des großen Bruders MS-DOS. Grundsätzlich: Eine Disk teilt sich ein in *Spuren* (*Tracks*) und *Sektoren*. Bei der SF354 sind es 80 Tracks, bei der SF314 sogar 2 x 80 Tracks, weil da die Disketten beidseitig beschrieben werden. Jeder Track besteht wiederum aus 9 Sektoren:



(Bild 1: So sieht der Lesekopf eine typische ST-Diskette)

Spur 79 liegt innen, Spur 0 außen. Zwischen den Sektoren gibt es kleine Lücken, auf die ich noch zu sprechen komme. In den Sektoren werden die eigentlichen Daten abgespeichert, beim ST 512 Bytes pro Sektor.

Information durch Rotation

Über der rotierenden Diskette schwebt ein Schreib-Lesekopf. Beim Schreiben fließt Strom durch den Schreibkopf und erzeugt eine magnetische Fährte auf der Diskette. Beim Lesen ist es umgekehrt: Die magnetisierten Partikel auf der Disk induzieren einen Stromfluß im Lesekopf, den er als logisch '1' oder '0' versteht. Aber keine Angst: So tief in die Physik brauchen wir nicht einzusteigen, der ST verfügt über einen Baustein (nämlich einen *Diskcontroller*), der sich um diese Dinge kümmert. Dieser Baustein im ST ist auch der Grund, warum die Programme dieses Kurses auch auf Fremdlaufwerken laufen: Die Floppystationen am ST sind nicht wie die anderen Rechner (C64) *intelligent*, sondern bestehen fast nur aus Mechanik.

Das bedeutet, die Ansteuerung hängt nur vom Controller ab, und der ist ja bei allen ST-Rechnern gleich.

Platz da!

Nun sieht man schon auf der Zeichnung, daß nach dem neunten Sektor noch viiiiel Platz ist, so viel Platz, daß viele Formatierprogramme (zum Beispiel) auch FATDISK aus der ST/86) dort einen zehnten Sektor unterbringen und so die Kapazität erhöhen. Das macht pr Spur immerhin 512 Bytes aus, bei 80 Spuren und 2 Seiten sind das 80 Kilobyte!

Bei vielen Laufwerken kann man auch den Schreib-Lesekopf bis über Track 79 hinaus bewegen. Einer eigenen Blitzumfrage bei Händlern und ST-Besitzern zufolge gibt es allerdings bei etwa der Hälfte aller Laufwerke Probleme ab Track 82, weil der Lesekopf anschlägt. Um herauszufinden, ob Ihr Laufwerk auch Track 82 noch mitmacht, sollten Sie mal eine freie Disk mit FATDISK formatieren. Legen Sie das Ohr aufs Laufwerk (ja, ich weiß, es klingt lächerlich, aber es lohnt sich) — wenn es bei Track 82 (FATDISK zeigt den aktuellen Track an) anders klingt als bei den vorherigen Trackwechseln (es hört sich etwa so an, als ob ein kleines Männchen in der Floppy den Lesekopf mit einem Hämmerchen bearbeitet), haben Sie Pech: Tyrannisieren Sie irgendjemanden, der Ihnen Ihre Floppy einstellen kann. Vielleicht kann der ja noch was machen. Andererseits können einige Leute aber auch ihren Kopf (quatsch, den ihrer

Floppystation) bis Spur 85 bewegen!

Ein wenig Arithmetik: 83 Spuren (also Spur 0 bis Spur 82) mal 10 Sektoren mal 2 Seiten macht 1660 Sektoren = 849920 Bytes. Von dieser Summe geht aber noch einiges für Bürokratie weg, bei FATDISK bleiben nur 828416 Bytes übrig.

Verwaltungskram auf Disketten

Da ist zum ersten der geheimnisumwitterte *Bootsektor* (Track 0, Sektor 1, Seite 0), der 512 Bytes frißt. Nach dem Einschalten prüft der ST den Bootsektor der Diskette in Laufwerk A. Wenn dieser Bootsektor *ausführbar* ist, versucht der ST von dieser Diskette ein TOS zu laden. Außerdem stehen noch viele andere Informationen im Bootsektor, die ich genauer in der zweiten Folge behandeln werde.

Ab dem zweiten Sektor steht die sogenannte *FAT* (*File Alloction Table*). Auf ihr ist die Belegung der Diskette mit Dateien vermekrt. Das Format ist allerdings a bisserl kompliziert. Vorweg: TOS verwaltet seine Disketten nicht in logischen Sektoren, sondern in *Clustern*. Das sind jeweils 2 aufeinanderfolgende Sektoren. Eine neue Datei belegt mindestens einen ganzen Cluster — selbst bei einer Datei mit nur einem Byte! Diese Verschwendungssucht schmiert aber den Verwaltungsmechanismus und beschleunigt die Bürokratie (Parallelen mit der ‚wirklichen' Welt sind rein zufällig). Eine typische FAT fängt nun so an:

```
        Eintrag für...
Kenn-      Cl. 2  Cl. 3 Cl. 4 Cl. 5
bytes
00 00 00 03 40 00 05 FF 0F ..
```

Jeder Cluster erhält in der FAT 12 Bit (!) zugewiesen, das sind drei Hexadezimal*ziffern*. Die ersten 5 Nullen sind Flags, deren Bedeutung ich noch nicht vollständig entschlüsselt habe. Bei MS-DOS, dem großen Bruder, steht im ersten Byte ein Formatkennzeichen, im zweiten ist die Kapazität der Floppy eingetragen. Seltsamerweise beeindrucken Änderungen der ersten beiden Bytes TOS nicht sonderlich.

Vielleicht hat sich ja schon ein Leser damit befaßt und scheibt mir oder der ST. Das obere Nibble des 3.Bytes bedeutet offenbar auch nichts Besonderes (wie gesagt, ich bin für Tips zugänglich). Aber dann folgt der Eintrag des Clusters mit der Nummer 2 (TOS speichert Daten erst ab Cluster 2 ab). Hier sehen wir '003', das heißt: Die Datei, die im Cluster 2 beginnt, wird im Cluster 3 fortgesetzt. Im Eintrag für Cluster 3 steht '400', also ein umgedrehtes '004' (warum auch immer), die Datei setzt sich folglich bei Cluster 4 fort undsoweiter.

Am Schluß der Datei steht ein Endekennzeichen (FFO). In MS-DOS zumindest gibt es noch die Kennzeichen FF7 (Sektor beschädigt) und FF8—FFF (Dateiende). Für jeden Cluster sind 12 Bit reserviert, macht (bei normalen Format) 8640 Bit = 1080 Byte, also knapp 3 Sektoren. Trotzdem reserviert ATARI großzügig 5 Sektoren für die FAT.

Tanz auf dem Vulkan

Direkt auf die erste FAT folgt eine identische Kopie der FAT, aus Sicherheitsgründen. Nett gedacht, nur: beide FATs liegen in der Regel auf der gleichen Spur, und da viele Programme gleich eine ganze Spur schreiben, segelt bei einem Fehlgriff auch die Kopie ins Nirwana. Traurig, traurig, denn nur mit diesem roten Faden (oder besser: FATen) kann man feststellen, welche Bereiche eine Datei belegt. Gerade bei zerstörten Dateien ist das sehr wichtig — und ohne FAT gleicht der Versuch, die Datei wieder zusammenzuklauben, einer stecknadelsuche im Vesuv.

Ein Tip für diejenigen, die es trotzdem versuchen wollen (oder müssen, mein Beileid): TOS versucht grundsätzlich, aufeinanderfolgende Sektoren für neue Dateien zu verwenden. Erst wenn es auf einen breits belegten Sektor stößt, weicht das Betriebssystem aus. Auf einer einigermaßen leeren Diskette hat man also noch relativ gute Chancen, seine Dateisektoren wiederzufinden.

Nach den FATs steht in 7 Sektoren die Directory, beim normalen Format ab dem 12. Sektor, vom Anfang der Diskette an gezählt (dabei zählt man zuerst die Sektoren der Vorderseite, dsann die der Rückseite, so vorhanden, und geht dann weiter zum nächsten Track). Jeweils 32 Bytes pro Datei beinhalten deren Namen und Kenndaten unter anderem auch einen Verweis auf den ersten Dateicluster. Damit gerüstet, schaut TOS an der betreffenden stelle der FAT nach und kann den Verlauf einer Datei verfolgen. Genaueres zur Directory im Teil 2 der Reise durchs wilde Floppystan (in der nächsten ST).

Kurze Überschlagsrechnung: Für die Verwaltung einer Diskette gehen summa summarum 18 Sektoren für Bootsektor, FATs und Direcotry hops, also 9216 Bytes.

Nochmal in der Zusammenfassung der Anfang einer ST-Diskette:

1. Sektor: Bootsektor
(Track 0, Seite 0, Sektor 1)
2. Sektor: FAT1 (Track 0, Seite 0, Sektor 2)
7. Sektor: FAT2 (Track 0, Seite 0, Sektor 7)
12. Sektor: Directory (bei SF314: Track 0, Seite 1, Sektor 3? bei SF 354: Track 1, Seite 0, Sektor 3)

Danach: Daten und Programme

Manche Formatierprogramme kürzen die beiden FATs noch auf jeweils 3 Sektoren (was völlig ausreicht) und die Directory auf 64 Einträge (also 4 Sektoren) und gewinnen so 7 Sektoren = 2584 Bytes.

Der Clou: HYPERFORMAT

Trotzdem, das alles kann noch nicht das Wahre sein: der AMIGA hat doch auch diselben Laufwerke und formatiert auf 880K. Warum kann das der ST nicht, fragt man sich. Er kann. Aber dazu mußte ich erst mal einiges umbiegen.

Glücklicher AMIGA, du hast es leichter. Das Betriebssystem des AMIGA formatiert nämlich eine Diskette mit 11 Sektoren pro Spur! Macht zusammen — bei 80 Spuren — über 900000 Bytes! Et voila, dachte ich mir, ds müßte der ST doch auch können, setzte mich fröhlich pfeifend an meinen ATARI-Ferrari, um die Sache auszuprobieren, und verlor schon nach kurzer Zeit die Lust: Ein schnell zusammengehacktes Formatierprogramm für 11 Sektoren formatierte zwar, doch überschrieb der elfte Sektor immer den ersten: Zuwenig Platz auf der Spur. Was tun? Irgendwo im Betriebssystem entdeckte ich schließlich die Formatierroute des XBIOS, die man auch über den internen XBIOS-Aufruf 10 erreicht. Wenn man analysiert, was TOS da so auf die Diskette schreibt, kommt man schließlich auf folgendes Track-Format: (siehe Tabelle 1)

Der lange Weg zum Glück

Das Problem: Es paßt zuwenig auf eine Spur, um einen elften Sektor schreiben zu können. Die Beobachtung: Zwischen den einzelnen Informationen auf der Spur liegen immer wieder Lückenbytes (*Gapbytes*). Schlußfolgerung: Wenn man die Lücken auf ein Minimum kürzt, bringt das vielleicht so viel Platz auf der Spur, daß noch ein elfter Sektor draufpaßt.

Und in der Tat: Man kann die Lückenbytes kürzen (durch Änderungen in der For-

```
Vorspann  (Index,  Trackheader;
"PoST Index Mark")

(*)  Lückenbytes  (Pre  Adress
Mark)
     Synchronisationsbytes
     Adressmarke + Sektorinfo
     Checksumme
      Lückenbytes  (PoST  Adress
Mark)
     Synchronisationsbytes
     Datenmarke + DATEN
     Checksumme
       Lückenbytes  (PoST  Data
Mark)
zurück zu (*), bis alle Sekto-
ren geschrieben sind, danach:

Lückenbytes bis zum Trackende.
```

(Tabelle 1: Einer Spur auf der Spur)

matierroutine des TOS), wenn man sich nur daran hält, daß mindestens 30 Lückenbytes zwischen Adreßmarke und Daten geschrieben werden. Das erste **HYPERFORMAT** modifiziert also zuerst mein RAM-TOS so, daß 11 Sektoren auf eine Spur passten, rief dann ganz normal das XBIOS auf, um zu formatieren, und gab dabei an, 11 Sektoren formatieren zu wollen.

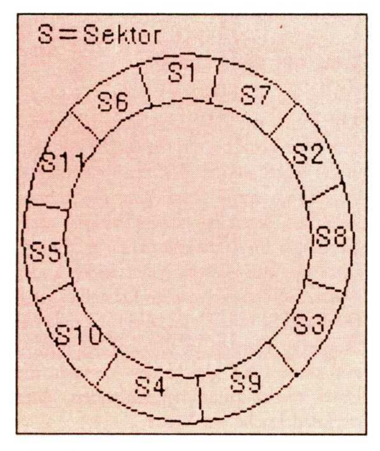
HYPERFORMATiert man eine Diskette nach diesem Verfahren, gibt es allerdings bald lange Gesichter:

1. Nicht jedes Laufwerk vertrug offenbar die Roßkur fürs XBIOS. Auf Laufwerken von Freunden ließen sich **HYPERFORMA**Tierte Disketten zunächst nicht lesen, später zwar lesen, aber nicht **HYPERFORMA**Tieren. Nach kniefieslingster Probiererei mit den Lückengrößen scheinen auch andere Laufwerke mein **HYPERFORMAT** zu verstehen, so daß ich damit an die öffentlichkeit treten kann, ohne sofortige Enthauptung zu fürchten. Sollte es doch nicht laufen, liegt das an physikalischen Toleranzen der Laufwerke.

Hexerei bringt Geschwindigkeit

2. Die Schreib-Lesegeschwindigkeit zwar zunächst ein Graus (3mal langsamer, als die sowieso nicht sehr schnelle SF314 sonst liest). Ein Freund kam auf die rettende Idee — Stichwort *Interleave* (dafür bekam er auch eine Widmung im Programm).

Normalerweise werden die Sektoren so auf die Spur geschrieben wie in Bild 1 angegeben, also erst Sektor 1, dann Sektor 2 und sofort. Beim normalen Format liest die Floppy zuerst den Sektor 1, der Floppycontroller (das ist der Baustein, der die Ansteuerung des Laufwerks übernimmt) werkelt jetzt ein bißchen, und wenn er wieder bereit ist, hat sich die Lücke zwischen den Sektoren gerade unterm Lesekopf vorbeigedreht, so daß gleich der nächste Sektor gelesen werden kann. Bei den gekürzten Lücken stimmt nun dieses Timing nicht mehr, und der Controller muß immer wieder Pausen einlegen und warten, bis der passende Sektor wieder unter dem Lesekopf vorbeirotiert. Nun kann man eine Spur aber auch so formatieren:



(Bild 2: Mehr Speed durch noch mehr Mut zur Lücke)

Damit hat der Controller zwischen zwei aufeinanderfolgenden Sektoren (etwa zwischen S1 und S2) immer genug Zeit! Diese Taktik nennt man *Interleaving*, der Interleavefaktor ist in diesem Falle — und auch per Voreinstellung in meinem Programm — sechs (weil auf die aktuelle Sektornummer immer sechs addiert wird; wenn das Ergebnis größer als die Anzahl der Sektoren pro Spur ist, wird diese Anzahl abgezogen). So modifiziert, bringt **HYPERFORMAT** eine Diskette auf beachtliche Geschwindigkeit: Im Durchschnitt ist sie im gemischten Schreib-Lesezugriff 20 bis 35 % schneller als normal formatierte Disketten, in Einzelfällen liegt dieser Wert sogar noch weit besser.

Dieser Geschwindigkeitsvorteil rührt auch daher, daß man beim Diskzugriff nicht mehr so oft den Track wechseln muß, weil mehr Daten auf eine Spur passen. Und der Trackwechsel ist auch eine zeitaufwendige Sace (normalerweise werden jeweils 3ms dafür verbraten).

3. In der bisherigen Version lief **HYPERFORMAT** nur auf RAM-TOS, da ja Passagen aus dem TOS geändert, *gepatcht* werden mußten. In der jetzigen Version ist auch das gelöst.

HYPERFORMAT prüft nach, ob ein ROM-TOS vorliegt, wenn nein, erwartet es ein RAM-TOS in der Version vom 6.2.86 im Speicher (es müßte auch mit der Version vom 18.7.85 laufen), wenn ja, benutzt es eine modifizierte ROM-Routine mit Einsprüngen ins ROM. Eine wichtige Anmerkung: Booten Sie ein FASTLOAD mit, wenn Sie den Geschwindigkeitsvorteil von **HYPERFORMAT**-Disketten nutzen wollen. Beim Umschreiben des Programmes auf ROM-Rechner habe ich bemerkt, daß das Timing beim Lesen und Schreiben nur mit FASTLOAD optimal ist. Deswegen installiert **HYPERFORMAT** bei gebootetem RAM-TOS gleich FASTLOAD und FASTFORMAT mit.

ROM-Beisitzer haben zwei Möglichkeiten: Ein FASTLOAD-Programm booten (in den AUTO-Ordner kopieren), das auch auf ROM-Rechnern läuft, oder die ROMs in Eproms kopieren und dabei auf den FASTLOAD-Modus umbrennen. Dazu ändert man im Eprom U7 den Inhalt der Eprom-Adresse $0DC7 von $14 auf $10 (modifizierte EPROMS gibt es auch für 30 Mark zu kaufen). Eine von diesen Möglichkeiten sollten Sie nutzen, um alle Vorteile von **HYPERFORMAT** genießen zu können. Die FASTLOAD-Modifikation läuft übrigens schon seit einem Jahr bei mir und bei vielen Bekannten ohne Kompliaktionen, also keine falsche Vorsicht.

Und ein kleines Trostpflaster für ST-Besitzer ohne FASTLOAD: Das Formatieren läuft bei **HYPERFORMAT** auch bei ROM-Rechnern mit derselben Geschwindigkeit, als wäre ein FASTLOAD mitgebootet worden! Und wenn Sie den Interleavefaktor auf 3 einstellen, ergibt sich sogar doch noch ein kleiner Geschwindigkeitsvorteil gegenüber normalen Disketten.

Allerdings werden Disketten mit Interleavefaktor 3 wieder langsam, wenn man sie mit FASTLOAD liest. Deswegen empfehle ich dringend (um auch vom Speed her die Kompatibilität zu wahren), den Interleave nicht zu verstellen, sondern ein FASTLOAD ins TOS zu integrieren.

Die Anleitung zu HYPERFORMAT

Das Programm ist vollständig in Assembler geschrieben und relokatibel; es wurde mit dem AS68 aus dem Entwicklungspaket assembliert. Andere Assembler brauchen eventuell andere Pseudo-Ops; bitte

halten Sie sich an die Anleitung Ihres Assemblers. Das Programm an sich ist recht lang zum Abtippen, seien Sie vorsichtig. Die Kommentare müssen Sie ja nicht treudoof runterhacken, aber lesen Sie die Hinweise, man kann dabei sicher was für die eigene Programmierung lernen. Nicht zuletzt deswegen ist **HYPERFORMAT** *heavily documented*, wie der Angelsachse so schön sagt.

Wenn Sie das Programm fertig eingetippt (ich beneide Sie nicht), assembliert (fluch!) und gelingt (bibber!) haben, dann starten und keine Bomben oder ATARI-Gardinen entdecken, haben Sie es geschafft, Sie sind im Menü angekommen.

Im Menü können Sie einige Parameter einstellen, zum Beispiel, wieviel Spuren Sie formatieren wollen (wichtig für Leute mit dem Knackatakatak bei der 82. Spur), wieviel Seiten Ihr Laufwerk hat (auf der einseitigen Floppy hat man dank **HYPERFORMAT** 459000 Bytes frei), wieviel Directoryeinträge Sie zulassen wollen, und welcher Interleavefaktor gewählt wird.

Voreingestellt sind 2 Seiten, 83 Tracks (also Spur 0 bis Spur 82 einschließlich), 64 Directoryeinträge und ein Interleavefaktor 6, und damit ergeben sich mehr als 927000 freie Bytes auf Ihrer Diskette (AMIGA go home!). **HYPERFORMAT** kürzt die FATs dabei auf völlig ausreichende drei Sektoren. CTRL-C im Menü bricht das Programm ab, RETURN startet das Formatieren, 'B' überspringt das Formatieren und schreibt nur Bootsektor, FAT und Directory neu. ROM-Besitzer ohne FASTLOAD können den Interleave auf 3 einstellen (siehe oben).

Zusätzlich sind im Listing noch folgende Parameter änderbar:

virgin (Daten, mit denen die Sektoren beim Formatieren initalisiert werden)
spt (Sektoren pro Track)
res (Reservierte Sektoren)
fat (Anzahl der FATs)
secsperfat (Länge der FAT)

Von allen anderen Parametern im Listing sollten Sie zunächst mal die Finger lassen.

Sollte Ihr Laufwerk doch mal bocken und Lesefehler melden, können Sie in den Routinen *change_tos* und *rom_fmt* die gekennzeichneten „Lücken" von 3 auf 4 vergrößern, das hilft meistens.

Diesmal haben Sie erfahren, wie eine Diskette aufgebaut ist, wie die Aufzeichnung funktioniert, was FATs, Cluster und Diskcontroller sind, wie man mit kleinen Tricks den Verwaltungsaufwand reduzeirt und mehr Platz gewinnt, und Sie haben jetzt mit **HYPERFORMAT** die ultimate Waffe gegen den Knetefraß durch Diskettenkauf. In der nächsten ST lesen Sie, wie man auf den Diskettencontroller des ST zugreift, was der für Finessen beherrscht, wie man BIOS, XBIOS und GEMDOS für die Arbeit mit der Floppy nutzen kann, wie gängige Kopierschutzmechanismen funktionieren und einiges mehr.

```
*************************************************************
* HYPERFORMATTER                                            *
* Written 1987 by Claus Brod                                *
*                 Am Felsenkeller 2                         *
*                 8772 Marktheidenfeld                      *
*                 09391/3206                                *
* (C) 1987 and for all eternity by Claus Brod               *
*                                                           *
* 1. Formatiert Diskette in Laufwerk A                      *
*    auf über 905K (>927000 Bytes frei)                     *
* 2. Beschleunigt Diskzugriff bei Laden                     *
*    und Speichern um 20-35%                                *
*                                                           *
* Version für RAM- und ROM-TOS                              *
* Revision 2.0, Last update 4.4.87                          *
* Assembliert mit AS68 aus dem Entwicklungspaket            *
*************************************************************

virgin      .equ $cbcb          * Virgin data
spt         .equ 11             * Sektoren pro Track
laufwerk    .equ 0              * Laufwerk A
secsperfat  .equ 3              * Sektoren pro FAT
spc         .equ 2              * Sectors per Cluster
res         .equ 1              * Reservierte Sektoren
fat         .equ 2              * Anzahl der FATs
bufflength  .equ 10000          * Länge des Trackpuffers


****************
* Routine zum Reservieren von Speicherplatz
****************

    move.l a7,a5              * Stackpointer retten
    move.l 4(a5),a5           * Basepage-Start vom Stack nach a5
    move.l $c(a5),d0          * Länge von .text
    add.l $14(a5),d0          * Länge von .data addieren
    add.l $1c(a5),d0          * Länge von .bss addieren
    add.l #$500,d0            * Platz für den Stack muß auch sein
    move.l d0,d1              * Ergebnis nach d1
    add.l a5,d1               * Ergebnis + Adresse der Basepage
    and.l #-2,d1              * Adresse gerade machen
    move.l d1,a7              * Stackpointer = errechnete Adresse
    move.l d0,-(sp)           * Größe des Bereiches
    move.l a5,-(sp)           * Startadresse, ab der reserviert wird
    clr.w -(sp)               * Dummy (reserviert)
    move.w #$4a,-(sp)         * SETBLOCK
    trap #1                   * im GEMDOS
    add.l #12,sp

    clr.l -(sp)               * Userstack wird Supervisorstack
    move.w #$20,-(sp)         * SUPER
    trap #1                   * im GEMDOS
    add.l #6,sp               * ab jetzt alles im Supervisormode
    lea save_ssp(pc),a2
    move.l d0,(a2)            * Supervisorstackpointer retten
```

(Tabelle 2: Hier noch eine Liste der Routinen von HYPERFORMAT:

main	Hauptschleife
change_tos	modifiziert RAM-TOS und inSTalliert FASTLOAD+FASTFORMAT
repair_tos	macht Änderungen am RAM-TOS rückgängig, nicht aber FASTLOAD und FASTFORMAT
formatdisk	ganze Diskette formatieren
formattrack	eine Spur formatieren
rom	Vorbereitungen für die romfmt-Routine
romfmt	Modfizierte Formatierroutine aus dem ROM
writeboot	Bootsektor und FATs initialisieren
writesector	einen Sektor schreiben
printmsg	STring ausgeben
waitforkey	Auf TaSTe warten
mallocerror, freeerror, fatal-error	Fehlermeldungen
parameters	Eingabeschleife fürs Menü
op_exe	Exe-Flag ändern
op_tracks	Anzahl der Tracks ändern
op_side	Anzahl der Seiten einSTellen
op_dir	Wieviele Directoryeinträge?
op_inter	Interleavefaktor ändern
calc_secs	Zahl der Sektoren auf der Disk ausrechnen

①

```
    cmpi.l #$00FC0000,$4F2          * von ROM gebootet?
    beq rom_ist_da                  * ja, weiter
    bsr change_tos                  * RAM-TOS patchen
    lea romflag(pc),a2
    move.b #-1,(a2)                 * Flag für 'kein ROM' setzen
rom_ist_da:
    bsr main                        * Ab ins Hauptprogramm

exitus:
    lea romflag(pc),a2
    cmpi.b #-1,(a2)                 * ROM drin?
    bne rom_sweet_rom               * jawoll, weiter
    bsr repair_tos                  * Heileheilegänschen fürs RAM-TOS
rom_sweet_rom:
    move.l save_ssp(pc),-(sp)       * zurück
    move.w #$20,-(sp)               * in den Usermode
    trap #1
    add.l #6,sp

    clr.w -(sp)                     * Raus in die boese Welt
    trap #1

**************************
* main: Holt Parameter, formatiert
* und gibt Gelegenheit zur Wiederholung
**************************

main:
    lea msg1(pc),a5                 * Intromeldung
    bsr printmsg                    * ausgeben

    bsr parameters                  * Parameter einstellen
    lea diskmsg(pc),a5
    bsr printmsg                    * 'Diskette einlegen'
    bsr waitforkey                  * auf Taste warten
    bsr calc_secs                   * Sektoren ausrechnen

    lea fmtflag(pc),a2
    cmpi.b #13,(a2)                 * Formatieren oder nur Bootsektor?
    bne nurboot
    bsr formatdisk                  * eigentliche Formatierroutine
nurboot:
    lea againmsg(pc),a5             * Noch'ne Message
    bsr printmsg

    bsr writeboot                   * Bootsektor etc. schreiben
    bsr getkey                      * auf Taste warten
    cmpi.b #'y',d0                  * nochmal?
    beq main                        * 'y' für ja, zurück
    cmpi.b #'Y',d0
    beq main                        * 'Y' für ja, zurück
    rts

*********************
* change_tos: ändert TOS ab
*********************
change_tos:
    lea     $000075c2,a5            * a5 auf Anfang der fmt-Routine
    move.b #3,$81(a5)               * Post Index Mark, evt. auf 4
    move.b #3,$8f(a5)               * Pre Adress Mark, evt auf 4
```

②

```
    move.b #21,$bf(a5)              * Post Adress Mark 1
    move.b #11,$cb(a5)              * Post Adress Mark 2
    move.b #3,$f9(a5)               * Post Data Mark, evt auf 4
    move.w #600,$118(a5)            * Nachspann
    move.l #$6000034c,$58(a5)       * FASTFORMAT
    suba.l a5,a5                    * a5 löschen
    move.b #$10,$7a1d(a5)           * FASTLOAD
    rts

*********************
* repair_tos: siehe oben, nur retour
*********************

repair_tos:
    lea     $000075c2,a5
    move.b #$3b,$81(a5)
    move.b #$b,$8f(a5)
    move.b #$15,$bf(a5)
    move.b #$b,$cb(a5)
    move.b #$27,$f9(a5)
    move.w #$578,$118(a5)
    rts

**********************
* formatdisk: von Track 0-endtrack formatieren
**********************

formatdisk:
    move.l #bufflength,-(sp)        * bufflength Bytes
    move.w #$48,-(sp)               * per MALLOC
    trap #1                         * (GEMDOS)
    addq.l #6,sp                    * reservieren
    tst.l d0                        * Fehler?
    bmi mallocerror                 * Rumpel...

    move.l d0,a6                    * Trackpufferadresse nach a6
    move.w #0,d4                    * ab Track 0 formatieren
floop:
    move.w #0,d5                    * Seite 0 formatieren
    bsr formattrack
    lea side(pc),a2
    cmpi.b #2,(a2)                  * 1 oder 2 Seiten?
    bne nureine                     * eine Seite, nächster Track
    move.w #1,d5                    * Seite 1 formatieren
    bsr formattrack
nureine:
    addq.w #1,d4                    * Nächster Track
    cmp.b endtrack(pc),d4           * alle Tracks formatiert?
    bls floop                       * Nein, weiter

    move.l a6,-(sp)                 * Reservierten Bereich per
    move.w #$49,-(sp)               * MFREE im GEMDOS
    trap #1                         * wieder freigeben
    addq.l #6,sp
    tst.l d0                        * Fehler?
    bne freeerror                   * Da soll doch...
    rts
```


SPREADSHEET
DATABASE
LOGiSTiX
FULLY INTEGRATED
GRAFOX
SPREADSHEET
DATABASE
LOGiSTiX
FULLY INTEGRATED
TIME PLANNING
GRAPHICS
LOGiSTiX
GRAFOX

(3)

```
************************
* formattrack: Einen Track formatieren
* d5:Seite, d4:Track, a6:Puffer
************************

formattrack:
    move.w #virgin,-(sp)          * Virgin
    move.l #$87654321,-(sp)       * Magic Number
    move.w ileave(pc),-(sp)       * Interleave
    move.w d5,-(sp)               * Seite
    move.w d4,-(sp)               * Track
    move.w #spt,-(sp)             * Sektoren pro Track
    move.w #laufwerk,-(sp)        * Laufwerk (A)
    clr.l -(sp)
    move.l a6,-(sp)               * Pufferadresse
    lea romflag(pc),a2
    move.b (a2),d0
    cmp.w #-1,d0                  * ROMs eingebaut?
    bne rom                       * ja, in ROM-Routine
    move.w #10,-(sp)              * Und ab ins XBIOS: flopfmt
    trap #14
    add.l #2,sp
redo_stk:
    add.l #24,sp
    tst d0                        * Fehler passiert?
    bmi fatalerror                * Oje oje
    rts

rom:
    move.l $4a2,a1
    movem.l d3-d7/a3-a7,-(a1)     * Register retten
    move.l a1,$4a2
    clr.l a5                      * a5 löschen
    lea $FC0000,a0                * Offset-Register für ROM-Calls
    bsr romfmt
    move.l $4a2,a1
    movem.l (a1)+,d3-d7/a3-a7     * Register holen
    move.l a1,$4a2
    bra redo_stk


*************************
* romfmt: modifzierte Formatierroutine des ROM
*************************

romfmt:
    jsr $1CBE(a0)                   * Disketten gewechselt?
    moveq.l #$FF,d0
    jsr $1A34(a0)                   * Parameter setzen
    lea $fc0000,a0
    jsr $1C14(a0)                   * select drive and side
    move.w $E(a7),$9D4(a5)          * Sektoren pro Track
    move.w $14(a7),$9D6(a5)         * Interleave
    move.w $1A(a7),$9D8(a5)         * Virgin
    moveq.l #2,d0
    lea $fc0000,a0
    jsr $1CF6(a0)                   * Diskette gewechselt
    lea $fc0000,a0
    jsr $1B28(a0)                   * Track anfahren
    bne fatalerror                  * Fehler beim Tracksuchen
```

(4)

```
    move.w $9C4(a5),$0(a1)         * aktueller Track
    move.w #$FFFF,$9E0(a5)
    bsr machhin                    * Track formatieren
    bne fatalerror                 * Fehler passiert

    move.w $9D4(a5),$9CA(a5)       * Sektoren/Track als Zähler
    move.w #1,$9C6(a5)             * mit Sektor 1 anfangen
    lea $fc0000,a0
    jmp $1ada(a0)                  * flopok (verify überspringen)

machhin:
    move.w #$FFF6,$9DE(a5)
    move.w #1,d3                   * mit Sektor 1 anfangen
    move.l $9cc(a5),a2             * Adresse des Puffers

    move.w #3,d1                   * 4 Lückenbytes  (evt. auf
    move.b #$4E,d0                 *    move.w #4,d1 ändern)
    bsr wmult                      * in Puffer schreiben
nsect2:
    move.w d3,d4                   * Sektornummer retten

nsect:
    move.w #3,d1                   * 4 Lückenbytes  (evt. auf
    clr.b d0                       *    move.w #4,d1 ändern)
    bsr wmult
    move.w #2,d1                   * 3 Syncbytes (nicht ändern!)
    move.b #$F5,d0                 * $F5 schreiben
    bsr wmult

    move.b #$FE,(a2)+              * $FE, Adreßmarke schreiben
    move.b $9C5,(a2)+              * Tracknr. schreiben
    move.b $9C9,(a2)+              * Seite
    move.b d4,(a2)+                * Sektor
    move.b #2,(a2)+                * Bytes pro Sektor, Hibyte
    move.b #$F7,(a2)+              * Checksumme schreiben

    move.w #$15,d1                 * 22 Lückenbytes (nicht ändern!)
    move.b #$4E,d0                 * $4E schreiben
    bsr wmult
    move.w #$b,d1                  * 12 Lückenbytes (nicht ändern!)
    clr.b d0                       * 0 schreiben
    bsr wmult
    move.w #2,d1                   * 3 Syncbytes (nicht ändern!)
    move.b #$F5,d0                 * $F5 schreiben
    bsr wmult
    move.b #$FB,(a2)+              * $FB, Datenblockmarke

    move.w #$FF,d1                 * 256 Worte
wrvirgin:
    move.b $9D8(a5),(a2)+          * Virgin data
    move.b $9D9(a5),(a2)+
    dbra d1,wrvirgin

    move.b #$F7,(a2)+              * Checksum schreiben
    move.w #3,d1                   * 4 Lückenbytes (evt. auf
    move.b #$4E,d0                 *    move.w #4,d1 ändern)
    bsr wmult
```

(5)

```
**************************
* writeboot: Bootsektor etc. schreiben
**************************

writeboot:
    lea bootsector(pc),a5          * Bootsektor-Adresse
    move.w executable(pc),-(sp)    * Ausführbarkeitsflag
    move.w #-1,-(sp)               * Disktyp nicht ändern
    move.l #$11000000,-(sp)        * Serial number zufällig erzeugen
    move.l a5,-(sp)                * Adresse des Puffers übergeben
    move.w #18,-(sp)               * PROTOBT
    trap #14                       * im XBIOS
    add.l #14,sp

    move.w #0,d5                   * Track 0
    move.w #1,d6                   * Sektor 1
    move.w #0,d4                   * Seite 0
    bsr writesector                * Bootsektor schreiben

    move.w #2,d6                   * Ab Sektor 2
    move.w #0,d4                   * Seite 0
    move.w #0,d5                   * Track 0
    lea cleansector(pc),a5         * cleansector schreiben
cnochmal:
    bsr writesector
    addq.w #1,d6
    cmp.w #12,d6
    bne cnochmal
    rts

*******************
* writesector: Einen Sektor schreiben
* Seite in d4, Track in d5, Sektor in d6, Pufferadresse in a5
*******************

writesector:
    move.w #1,-(sp)                * Count
    move.w d4,-(sp)                * Seite
    move.w d5,-(sp)                * Track
    move.w d6,-(sp)                * Sektor
    move.w #laufwerk,-(sp)         * Laufwerk
    clr.l -(sp)                    * Füllsel
    move.l a5,-(sp)                * Pufferadresse
    move.w #9,-(sp)                * FLOPWR
    trap #14                       * Xbios
    add.l #20,sp
    rts

*******************
* printmsg: String ausgeben
* Adresse in a5
*******************

printmsg:
    move.l a5,-(sp)                * Message ausgeben
    move.w #9,-(sp)
    trap #1                        * GEMDOS
    addq.l #6,sp
    rts
```

(6)

```
    add.w $9D6(a5),d4              * Interleave draufzählen
    cmp.w $9D4(a5),d4              * mit größter Sektornummer
    ble nsect                      * vergleichen

    addq.w #1,d3                   * Startsektor plus eins
    cmp.w $9d6(a5),d3              * Interleave
    ble nsect2

    move.w #600,d1                 * 601 Lückenbytes (Änderung
    move.b #$4E,d0                 * zwecklos)
    bsr wmult

    move.b $9CF(a5),$FFFF860D      * DMA-Lowbyte
    move.b $9CE(a5),$FFFF860B      * DMA-Midbyte
    move.b $9CD(a5),$FFFF8609      * DMA-Highbyte
    move.w #$190,(a6)              * DMA-Status säubern
    move.w #$90,(a6)
    move.w #$190,(a6)              * Datenrichtung
    move.w #$1F,d7
    lea $fc0000,a0
    jsr $1C90(a0)                  * d7 an Controller schicken

    move.w #$180,(a6)
    move.w #$F0,d7                 * format_track-Befehl
    lea $fc0000,a0
    jsr $1C90(a0)                  * d7 an Controller

    move.l #$40000,d7              * timeout-Zähler

mfploop:
    btst #5,$fffffa01              * Controller fertig?
    beq dmastatus                  * ja, weiter
    subq.l #1,d7                   * nein, warten
    bne mfploop

    lea $fc0000,a0
    jsr $1BFA(a0)                  * abbrechen
kaputt:
    moveq.l #1,d7                  * Fehler
rts

dmastatus:
    move.w #$190,(a6)              * DMA-Status
    move.w (a6),d0                 * lesen
    btst #0,d0                     * Fehler?
    beq kaputt                     * ja, Fehler

    move.w #$180,(a6)
    lea $fc0000,a0
    jsr $1CA4(a0)                  * Register lesen
    lea $fc0000,a0
    jsr $165A(a0)                  * Fehlernummer
    and.b #$44,d0
    rts

wmult:                             * Bytes in Puffer schreiben
    move.b d0,(a2)+
    dbra d1,wmult
    rts
```

7

```
**********************
* waitforkey: 'Taste drücken' ausgeben
* und auf Taste warten
**********************

waitforkey:
   lea keymsg(pc),a5          * 'Taste drücken'
   bsr printmsg

getkey:                       * Einsprungpunkt ohne Message
   move.w #7,-(sp)            * Auf Taste warten
   trap #1                    * GEMDOS Conin without echo
   addq.l #2,sp
   rts

***********************
* Error-Routine mit Einsprungpunkten für verschiedene
* Fehlermeldungen. Gibt Fehler aus und verabschiedet sich
***********************
mallocerror:
   lea mallocmsg(pc),a5       * Fehler beim Reservieren von Speicher
   bra error

freeerror:
   lea freemsg(pc),a5         * Fehler bei der Freigabe von Speicher
   bra error

fatalerror:
   lea errormsg(pc),a5        * Fehler beim Formatieren

error:
   bsr printmsg               * Error-Routine, Textadresse in a5
   bsr waitforkey             * Gibt Message aus, wartet auf Taste
   bra exitus                 * und terminiert

**************************
* parameters: Gibt Menue aus
* und erlaubt Änderungen
**************************

parameters:
   lea msg_parms(pc),a5
   bsr printmsg               * Menue ausgeben

   bsr getkey                 * Auf Taste warten
   cmpi.b #3,d0               * ^C?
   beq exitus                 * ja, raus
   cmpi.b #13,d0              * RETURN?
   beq end_parms              * ja, fertig
   cmpi.b #'b',d0             * 'b' für boot
   beq end_parms
   cmpi.b #'B',d0
   beq end_parms

   cmpi.b #'d',d0             * 'd' gedrückt
   beq op_dir                 * Dir-Einträge ändern
   cmpi.b #'D',d0
   beq op_dir
   cmpi.b #'s',d0             * 's' gedrückt
   beq op_side                * Seitenzahl ändern
   cmpi.b #'S',d0
   beq op_side
   cmpi.b #'t',d0               * 't' gedrückt
   beq op_tracks                * letzten Track ändern
   cmpi.b #'T',d0
   beq op_tracks
```

8

```
   cmpi.b #'e',d0             * 'e' gedrückt
   beq op_exe                 * Ausführbarkeit ändern
   cmpi.b #'E',d0
   beq op_exe
   cmpi.b #'i',d0             * 'i' gedrückt
   beq op_inter               * Interleave ändern
   cmpi.b #'I',d0
   beq op_inter
   bra parameters

end_parms:
   lea fmtflag(pc),a2         * fmt oder nur boot
   move.b d0,(a2)
   rts

op_exe:                       * Ausführbarkeitsflag ändern
   lea s_exe(pc),a2           * Stringadresse
   lea executable(pc),a3      * Adresse der Variablen
   move.w (a3),d0
   move.b #'y',d1             * Yes als Default
   eor.w #1,d0                * Flag invertieren
   bne wr_exe                 * nicht ausführbar?
   move.b #'n',d1             * auf 'No' ändern
wr_exe:
   move.b d1,(a2)             * String und
   move.w d0,(a3)             * Flag ändern
   bra parameters

op_tracks:                    * Tracks ändern
   lea s_tracks(pc),a2        * Stringadresse
   lea endtrack(pc),a3        * Adresse der Variablen
   move.b $1(a2),d0           * Low-Nibble des Strings
   move.b (a3),d1             * Wert
   addq.b #1,d1               * um eins erhöhen
   addq.b #1,d0
   cmpi.b #'7',d0             * mehr als 86 Tracks läuft nicht
   bne wr_tracks
   move.b #'0',d0
   move.b #79,d1              * letzter Track 79

wr_tracks:
   move.b d0,$1(a2)           * String und
   move.b d1,(a3)             * Flag ändern
   bra parameters

op_side:                      * Seiten ändern
   lea s_sides(pc),a2         * Stringadresse
   lea side(pc),a3            * Adresse im Bootsektor
   move.b (a3),d0             * Seitenzahl nach d0 holen
   move.b $1(a2),d1
   eor.b #3,d0                * Seitenzahl ändern
   eor.b #3,d1
   move.b d0,(a3)             * in Bootsektor schreiben
   move.b d1,$1(a2)           * für String aufbereiten
   bra parameters
op_dir:
   lea s_dir(pc),a2           * Stringadresse
   lea entries(pc),a3         * Adresse der Einträge
   move.b (a3),d0
   move.b (a2),d1
```

9

```
    addq.b #1,d1
    add.b #16,d0                * 16 aufaddieren
    cmpi.b #143,d0              * schon mehr als erlaubt?
    bne wr_dir                  * nein, fertig
    move.b #15,d0               * wieder von vorne
    move.b #'1',d1
wr_dir:
    move.b d0,(a3)              * Bootsektor aktualisieren
    move.b d1,(a2)              * String aktualisieren
    bra parameters

op_inter:
    lea s_inter(pc),a2          * Stringadresse
    lea ileave(pc),a3           * Interleave-Adresse
    move.w (a3),d0              * Wert holen
    move.b 1(a2),d1             * String holen
    addq.b #1,d1
    addq.b #1,d0
    cmpi.b #10,d0               * schon 10?
    bne wr_inter
    move.w #1,d0
    move.b #'1',d1
wr_inter:
    move.b d1,1(a2)
    move.w d0,(a3)
    bra parameters

****************************
* calc_secs: Anzahl der Sektoren ausrechnen
* Formel: (endtrack+1)*11*side
****************************
calc_secs:
    clr.l d1
    clr.l d0
    lea sectors(pc),a3
    move.b endtrack(pc),d0      * Letzter Track
    addq.b #1,d0                * plus eins
    mulu #11,d0                 * mal 11
    move.b side(pc),d1
    mulu d1,d0                  * mal side
    move.b d0,(a3)              * Lowbyte schreiben
    asr.l #8,d0                 * Highbyte holen
    move.b d0,$1(a3)            * und schreiben
    rts

****************************
* Der Text
****************************
.even
msg1:
.dc.b 27,'E',27,'p                                          ',13,10
.dc.b 'T H E  H Y P E R  F O R M A T T E R  V2.0',13,10
.dc.b '=========================================',13,10
.dc.b '        Written 1987 by Claus Brod       ',13,10
.dc.b '       (Thanks to CD and Anton)          ',27,'q',13,
.dc.b '    927000 Bytes free & 30% faster',0

msg_parms:
.dc.b 27,'Y',32+7,32+0
.dc.b 27,'pT',27,'q  Tracks
s_tracks:
```

10

```
.dc.b '83',13,10
.dc.b 27,'pE',27,'q  Ausführbar (executable):'
s_exe:
.dc.b 'n',13,10
.dc.b 27,'pD',27,'q  Directoryeinträge       :$'
s_dir:
.dc.b '40',13,10
.dc.b 27,'pS',27,'q  Seiten (sides)          :'
s_sides:
.dc.b '02',13,10
.dc.b 27,'pI',27,'q  Interleave              :'
s_inter:
.dc.b '06',13,10
.dc.b 13,10,27,'pCR',27,'q HYPERFORMAT ',13,10
.dc.b 27,'p^C',27,'q Quit',13,10
.dc.b 27,'pB ',27,'q Bootsektor etc. schreiben',13,10,0

.even
againmsg:
.dc.b 13,10,13,10,'Again (Y/N)? ',0
diskmsg:
.dc.b 13,10,'Bitte Disk in Laufwerk ',65+laufwerk,' einlegen.',13,10,0
keymsg:
.dc.b 'Taste drücken.',13,10,0
errormsg:
.dc.b 13,10,'Fehler beim Formatieren!',13,10,0
mallocmsg:
.dc.b 13,10,'MALLOC Error!',13,10,0
freemsg:
.dc.b 13,10,'MFREE Error!',13,10,0

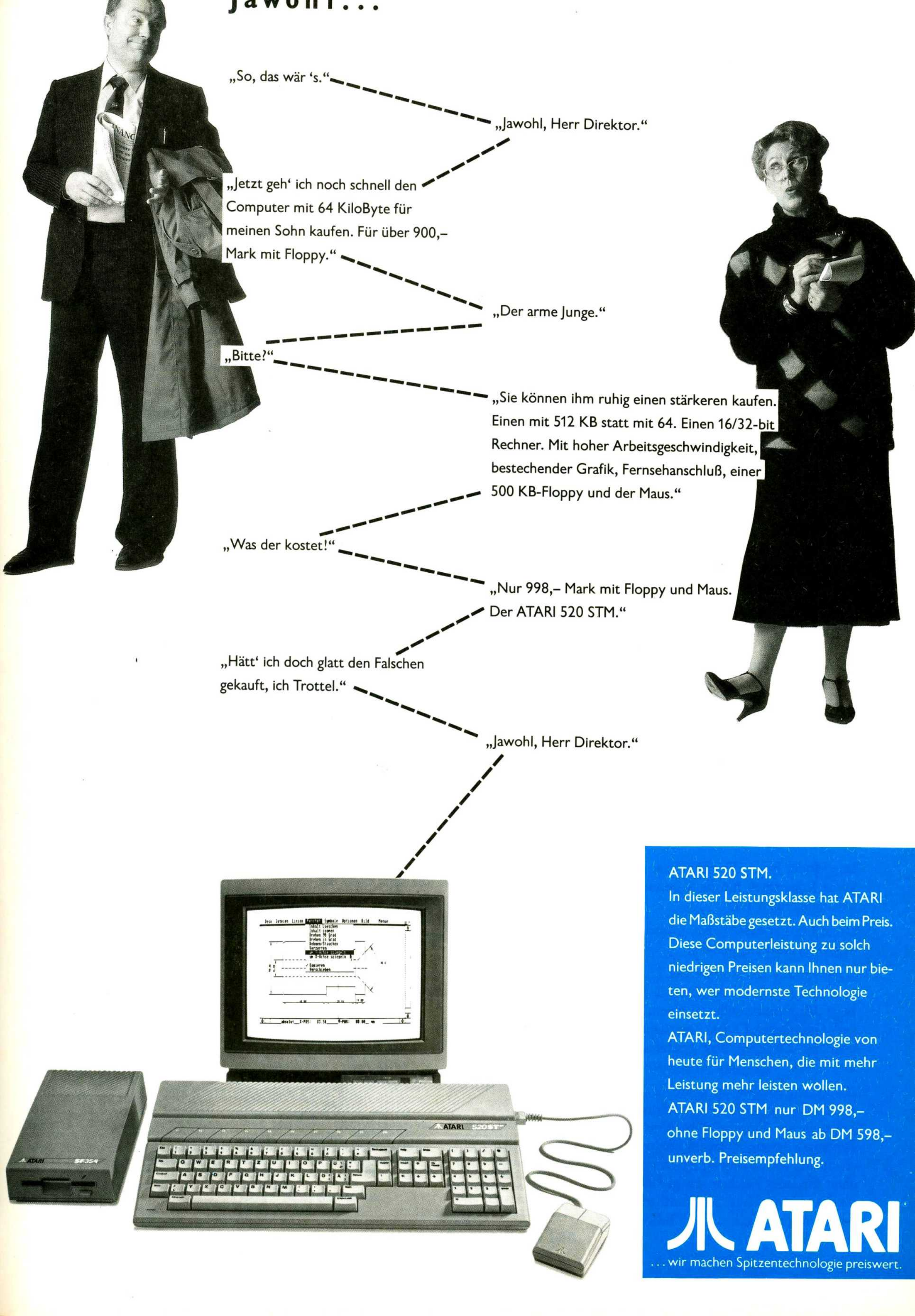
.even
bootsector:
.dc.b 96,56,76,111,97,100,101,114
.dc.b 203,31,238,0,2,spc,res,0,fat
entries:
.dc.b 63,0
sectors:
.dc.b 34,7,249,secsperfat,0,spt,0
side:
.dc.b 2,0,0,0,0,0,0,0,0,0,0,0,0,4
.dc.b 0,0,0,0,128,0,84,79
.dc.b 83,32,32,32,32,32,73,77
.dc.b 71,0,51,250,255,226,0,0
.dc.b 4,130,63,57,0,0,4,70
.dc.b 63,60,0,7,78,77,88,79
.dc.b 74,128,103,0,0,246,42,64
.dc.b 65,250,255,208,74,144,102,6
.dc.b 32,185,0,0,4,50,48,45
.dc.b 0,8,225,72,208,128,56,64
.dc.b 217,250,255,184,48,58,255,170
.dc.b 103,16,60,58,255,166,56,58
.dc.b 255,164,38,122,255,162,96,0
.dc.b 0,180,60,45,0,10,56,45
.dc.b 0,8,216,109,0,6,38,122
.dc.b 255,146,97,0,0,178,102,0
.dc.b 0,170,32,76,48,45,0,6
.dc.b 225,72,227,72,65,240,0,0
.dc.b 67,250,255,124,144,252,0,32
.dc.b 177,204,109,0,0,142,112,10
.dc.b 18,48,0,0,178,49,0,0
.dc.b 102,234,81,200,255,244,126,0
.dc.b 30,40,0,27,225,79,30,40
.dc.b 0,26,44,122,255,78,38,122
.dc.b 255,70,66,132,190,124,15,240
```

```
.dc.b 108,82,54,7,85,67,198,237
.dc.b 0,2,214,109,0,12,184,124
.dc.b 0,64,108,8,74,68,103,14
.dc.b 182,69,103,16,97,72,102,66
.dc.b 225,140,227,140,215,196,60,3
.dc.b 58,3,66,132,216,109,0,2
.dc.b 218,109,0,2,52,7,226,74
.dc.b 212,71,18,54,32,1,225,73
.dc.b 18,54,32,0,8,7,0,0
.dc.b 103,2,232,73,194,124,15,255
.dc.b 62,1,96,168,74,68,103,4
.dc.b 97,12,102,6,47,58,254,224
.dc.b 78,117,66,128,78,117,63,57
.dc.b 0,0,4,70,63,6,63,4
.dc.b 47,11,66,103,63,60,0,4
.dc.b 78,77,222,252,0,14,74,64
.dc.b 78,117,78,101,117,116,101,114
.dc.b 32,66,111,111,116,101,114,13
.dc.b 10,40,67,41,49,57,56,53
.dc.b 32,65,116,97,114,105,32,67
.dc.b 111,114,112,46,13,10,0,0
.dc.l 0,0,0,0,0,0,0,0
.dc.l 0,0,0,0,0,0,0,0
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0

cleansector:
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0     * Je 16 Nuller
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
.dc.l 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

ileave:
.dc.w 6
endtrack:
.dc.b 82
executable:
.dc.w 0
fmtflag:
.dc.b 0
romflag:
.dc.b 0
.even
save_ssp:
.dc.l 0
.end
```

(11)


Jawohl...
„So, das wär 's."
„Jawohl, Herr Direktor."
„Jetzt geh' ich noch schnell den Computer mit 64 KiloByte für meinen Sohn kaufen. Für über 900,– Mark mit Floppy."
„Der arme Junge."
„Bitte?"
„Sie können ihm ruhig einen stärkeren kaufen. Einen mit 512 KB statt mit 64. Einen 16/32-bit Rechner. Mit hoher Arbeitsgeschwindigkeit, bestechender Grafik, Fernsehanschluß, einer 500 KB-Floppy und der Maus."
„Was der kostet!"
„Nur 998,– Mark mit Floppy und Maus. Der ATARI 520 STM."
„Hätt' ich doch glatt den Falschen gekauft, ich Trottel."
„Jawohl, Herr Direktor."
ATARI 520 STM.
In dieser Leistungsklasse hat ATARI die Maßstäbe gesetzt. Auch beim Preis. Diese Computerleistung zu solch niedrigen Preisen kann Ihnen nur bieten, wer modernste Technologie einsetzt.
ATARI, Computertechnologie von heute für Menschen, die mit mehr Leistung mehr leisten wollen.
ATARI 520 STM nur DM 998,– ohne Floppy und Maus ab DM 598,– unverb. Preisempfehlung.
ATARI
...wir machen Spitzentechnologie preiswert.
ATARI SF354
ATARI 520ST

Komfort in Dialogboxen mit Modula-2

Die normalen Editierroutinen für Dialogboxen bieten nicht viele Möglichkeiten. Bei vielen Anwendungen könnte man speziellere Routinen benötigen. Wie wär's z.B. mit Fett- oder Kursivschrift in Dialogboxen?

Auslöser dieser Routinen war ein kleines Programm in Modula-2, das einen in einer Dialogbox eingegebenen Text sofort an den Drucker schickt; ihn also als Schreibmaschine benutzt. Das Problem war die Eingabe verschiedener Schriftarten, die natürlich benutzt werden sollten. Die normale GEM-Routine ‚FormDo' bietet solche Möglichkeiten nicht, also mußte eine entsprechende Prozedur geschrieben werden. Das Ergebnis zeigt *Bild 1*.

Die Routine ‚ObjectEditExtended' bietet diese Editiermöglichkeiten. Die vorhandene Tastenbelegung ist in *Bild 2* aufgelistet. Das Programm selber ist in Modula-2 geschrieben und in *Listing 1* abgedruckt.

Benutzen ...

Das Modul bietet die eigentliche Routine sowie Konstanten für die Schriftarten. Sie wird aufgerufen mit

ObjectEditExtended (Tree, Object, Typelength, typeline, typestyle);

‚Tree' enthält die Adresse des Dialogbaumes, in dem editiert werden soll. Die Anzeige der Eingabe sollte in einem normalen GraphicBox-Objekt geschehen, bei dem das ‚Outlined'-Flag gesetzt ist. Sein Index wird in ‚Object' übergeben.

‚Typenlength' gibt an, wieviele Zeichen maximal eingegeben werden können. Das Ergebnis der Eingabe steht hinterher in ‚typeline', das ein einfaches Feld von Zeichen, also ein String ist. ‚typestyle' gibt dann für jedes Zeichen mit gleichem Feldindex die gewählte Schriftart an. Diese wird entsprechend der VDI-Kodierung dargestellt.



Eine Überprüfung, ob die bereitgestellten Felder von der Länge her ausreichen, findet nicht statt, könnte aber eingebaut werden. Ebenso ist es nicht möglich, einen schon vorhandenen String editieren zu lassen, da ‚typline' immer zunächst mit Blanks aufgefüllt wird (die auch nach der Editierung eventuell hinter der Eingabe noch vorhanden sind). Wer will, kann die Routine also noch verfeinern.

Eine kleine Demonstration der Benutzung von ‚ObjectEditExtended' findet sich in *Listing 2*. Die dabei verwendete einfache Dialogbox-Resource (siehe *Bild 1*) wurde mit dem Programm ‚RSCMAKER' aus dem Modula-2 Toolkit umgewandelt.

... und dazulernen

Da die einzelnen Programmschritte im Listing kommentiert sind, wollen wir nun

```
Pfeil links     Cursor nach links bewegen
Pfeil rechts    Cursor nach rechts bewegen
Delete          Zeichen unter dem Crusor löschen und Rest der
                Zeile nach links ziehen
Backspace       Zeichen links vom Cursor löschen und Rest der
                Zeile sowie den Cursor nach links ziehen
Esc             Zeile löschen
Return          Editieren beenden
Enter           Editieren beenden
F1              Unterstreichung ein-/ausschalten
F2              Kursivschrift  ein-/ausschalten
F3              Hellschrift ein-/ausschalten
```

Bild 2

ATARI-BESTSELLER



CAD ist mehr als „nur" Computergrafik. Neben den speziellen Programmiertechniken und den typischen CAD-Prozeduren braucht man noch solide Grundkenntnisse zum Aufbau eines CAD-Systems. Wissen, das in diesem Buch kompakt und leichtverständlich zusammengefaßt wurde. Zudem können Sie anhand der einzelnen, beschriebenen Module leicht ein komplettes CAD-System erstellen. So wird das Gelernte gleich in die Praxis umgesetzt.
ATARI ST – Einführung in CAD
Hardcover, ca. 300 S. inkl. Disk., DM 69,–



Sie suchen einen praxisorientierten, einfachen Einstieg in die Programmiersprache C? Hier ist er: C für Einsteiger. Doch beschränkt sich dieses Buch nicht nur auf die grundsätzliche Bedienung und Struktur von C, sondern vermittelt auch echtes Profi-Wissen. So z.B. zur GEM- und Fensterprogrammierung. Mit den zahlreichen Tips und Tricks zur C-Programmierung formt dieses Buch aus jedem Einsteiger einen Profi.
C für Einsteiger
ca. 300 S., DM 39,–



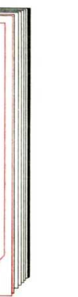
Die Grafikfähigkeit des ST gezielt für eigene Anwendungen einsetzen – dieses Buch zeigt Ihnen, wie es geht. Angefangen von den Grundlagen des VDI, GEM, AES und TOS bis hin zu speziellen Problemlösungen wie Programmierung des Rasterinterrupts oder einer flackerfreien Animation finden Sie hier alles zum Thema Grafik auf dem ST. Mit zahlreichen Utilitys in BASIC, C und Assembler.
Das Supergrafikbuch zum ATARI ST
Hardcover, ca. 600 S., inkl. Disk., DM 69,–



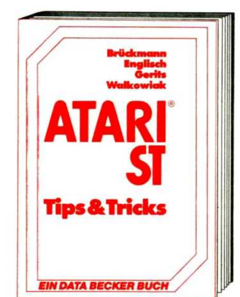
Neben der historischen Entwicklung und den verschiedenen Anwendungsgebieten Künstlicher Intelligenz lernen Sie hier auch gleich die Praxis kennen – auf Ihrem ST: Anhand intelligenter Denkspiele sowie dem Expertensystem „INFOMAT" werden Aufbau und Struktur von KI-Programmen erklärt. Selbstverständlich erfahren Sie auch alles Wissenswerte zu den typischen Programmiersprachen zur KI – LOGO, PROLOG und Lisp.
Einführung in die Künstliche Intelligenz
Hardcover, 406 S., DM 49,–



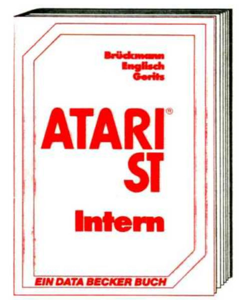
GFA-BASIC lernen einmal anders. Schritt für Schritt werden Ihnen anhand eines Supergrafikprogramms alle Befehle – auch die der Version 2.0 – erklärt und vermittelt. Mit all den nötigen Tips und Tricks, wie z.B. Rastertechnik, System Calls, Window- und Objektprogrammierung. Neben einem ausführlichen Einsteigerteil und einer kompletten Befehlsübersicht noch ein Bonbon: Eine detaillierte Beschreibung des GFA-Compilers.
Das große GFA-BASIC-Buch
Hardcover, 574 S., DM 49,–



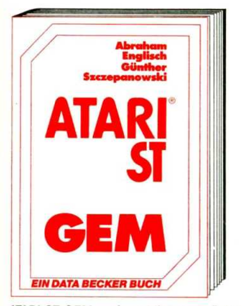
Der erste Buch zum ATARI ST ist für viele das wichtigste. Denn der richtige Einstieg garantiert später die volle Nutzung dieses Superrechners. ATARI ST für Einsteiger ist eine leichtverständliche Einführung in Handhabung, Einsatz und Programmierung Ihres ST: Von der Tastatur und der Maus hin zum Desktop bis zum ersten BASIC-Befehl und schließlich zu ganzen Programmen.
ATARI ST für Einsteiger
262 S., DM 29,–



Randvoll mit guten Ideen: Das ist unser Tips und Tricks zum ATARI ST. Sie suchen einen Druckerspooler? In diesem Buch finden Sie ihn. Sie brauchen eine Farb-Hardcopy-Routine? Bitteschön. Eine RAM-Disk anlegen? Nichts leichter als das. Viele neue Routinen, die problemlos in eigene Programme eingebunden werden können, warten nur darauf, daß Sie sie ausprobieren.
ATARI ST Tips & Tricks
Hardcover, 352 S., DM 49,–



Das Informationspaket zum ATARI ST. Geschnürt vom bewährten INTERN-Team Gerits, Englisch, Brückmann. Aus dem Inhalt: Der 68000-Prozessor, der I/O-Controller MFP 68901, der Soundchip YM-2149, alles über die Schnittstellen des ST und XBIOS, kommentiertes BIOS-Listing, wichtige Systemadressen, MIDI-Interface, Grafikaufbau – die Pflichtlektüre für ST-Profis.
ATARI ST Intern
Hardcover, 506 S., DM 69,–



ATARI ST GEM – ein schlichter, präziser Titel für ein Buch, das alles bietet, was es zu GEM zu sagen gibt: Systemaufrufe aus GFA-BASIC, C und Assembler, Erstellung eigener GEM-Bindings, Aufbau der Resourcen, Programmierung von Slider-Objekten, Aufbau eines eigenen Desktop und vor allen Dingen: ein komplett kommentiertes VDI-Listing sowie ein kommentiertes Listing ausgewählter AES-Funktionen.
ATARI ST GEM
Hardcover, 691 S., DM 69,–



Hier findet man Dinge, die in keinem Handbuch stehen. Sequentielle und RANDOM-ACCESS-Dateien, Controllerbeschreibung, Programmierung unter TOS, BIOS und XBIOS, Boot-Sektor, Programmformate auf Diskette und Festplatte, RAM-Disk und als Bonbon: Ein kompletter, vielseitiger Diskettenmonitor – eben alles, was dazugehört, wird hier mit vielen anschaulichen Beispielen erläutert.
ATARI ST Floppy und Harddisk
Hardcover, 522 S., DM 59,–



68000-Assembler – der Schlüssel zur Programmierung der spektakulärsten Effekte und der leistungsfähigsten Programme auf dem ATARI ST. Dieses Buch zeigt Ihnen, wie Sie Ihren ST mit Maschinensprache voll ausnutzen können: Verwendung von Systemroutinen, Bitmanipulationen, Rekursion, Stacks, Einbinden von Assemblerroutinen in Hochsprachen – vom ersten Schritt bis hin zum echten Einsatz von Maschinensprache.
ATARI ST Maschinensprache
334 S., DM 39,–

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name
Straße
Ort

ST 67

noch etwas allgemeiner die Arbeitsweise erläutern.

Die Routine muß einzelne Buchstaben, passsend für das angegebene Objekt, auf den Bildschirm schreiben. Sie muß dazu zunächst die Größe der Buchstaben erfahren, was mit ‚GrafHandle‘ geschieht. Beim Aufruf wird dann mit ‚ObjectOffset‘ die Lage des Objekts festgestellt. Mit diesen Angaben können wir die Buchstaben passend hineinschreiben.

In ‚cyrstyle‘ wird die momentan angewählte Schriftart gehalten. Die Funktionstasten verändern diesen Wert, der dem VDI mit ‚SetGraphicTextEffects‘ mitgeteilt wird und dann bei jedem ‚GraphicText‘-Aufruf verwendet wird.

Den Cursor erzeugen wir, indem ein Rechteck der Größe eines Zeichens mit ‚DrawBar‘ gezeichnet wird (‚SetCursor‘). Durch Setzen des Schreibmodus auf Revers (Parameter 3 bei ‚SetWritingMode‘) wird nicht wirklich gezeichnet; vielmehr invertiert der Aufruf genau einen Buchstaben, nämlich den unter dem Cursor. Ein erneuter Aufruf von ‚SetCursor‘ entfernt ihn wieder.

Interessant ist auch noch die Implementierung der Backspace- und Delete-Taste. Bei ihnen muß jeweils der Bereich rechts vom Cursor um ein Zeichen nach links verschoben werden. Außerdem geschieht die Zeicheneingabe immer im Einfügemodus; dann muß sich dieser Teil nach rechts bewegen.

Dazu dient ‚MoveLine‘, das zum Kopieren auf dem Bildschirm ein Rastercopy verwendet. Dafür notwendig ist ein Memory-Form-Definition-Block (MFDB), der. die Bildschirmadresse sowie eine Beschreibung der momentanen Auflösung enthält. Wir besorgen uns diese Informationen im Initialisierungsteil.

Die zu kopierenden Bildschirmbereiche lassen sich leicht mit der Zeichengröße sowie der Objektposition errechnen. Durch die schnelle Routine ‚CopyRasterOpaque‘ erhalten wir eine prompte und fließende Reaktion auf dem Bildschirm. Da die Prozedur Werte wie die Zeichengröße vom GEM abfragt, läuft sie übrigens problemlos in Farbe und Schwarz-Weiß.

Die Routine ist neben den oben genannten Verfeinerungen noch um weitere Sondertasten leicht erweiterungsfähig. Man könnte bei Bedarf auch noch die Schriftarten Outlined und Shadowed implementieren.

(RT)

```
DEFINITION MODULE AESExtender;

FROM SYSTEM IMPORT ADDRESS;

CONST NormalStyle = 0;
      Bold        = 1;
      Light       = 2;
      Italic      = 4;
      Underline   = 8;

PROCEDURE ObjectEditExtended(Tree:ADDRESS; Object, Typelength:INTEGER;
                             VAR typeline:ARRAY OF CHAR;
                             VAR typestyle:ARRAY OF INTEGER);

END AESExtender.
```

Listing 1a

```
IMPLEMENTATION MODULE AESExtender;                                    (1)

(*  Robert Tolksdorf - finished 8.4.87 *)

(* optimieren *)
(*$Q+*)

FROM SYSTEM           IMPORT ADR, ADDRESS ;
FROM XBIOS            IMPORT ScreenPhysicalBase, GetResolution,
                             Low, Medium, High ;
FROM GEMAESbase       IMPORT MouseOff, MouseOn, BeginMouseControl,
                             BeginUpdate, EndUpdate, EndMouseControl ;
FROM AESEvents        IMPORT EventKeyboard ;
FROM AESGraphics      IMPORT GrafMouse, GrafHandle ;
FROM AESObjects       IMPORT ObjectOffset ;
FROM AESWindows       IMPORT WindowUpdate ;
FROM GEMVDIbase       IMPORT BigPxyArrayType, PxyArrayType,
                             TextAttrArrayType ;
FROM VDIAttribs       IMPORT SetWritingMode, SetGraphicTextEffects,
                             SetGraphicTextColour, SetFillInteriorStyle,
                             SetFillColour, SetGraphicTextAlignment ;
FROM VDIInquires      IMPORT InquireTextAttributes,
                             InquireFillAttributes ;
FROM VDIOutputs       IMPORT GraphicText, DrawBar ;
FROM VDIRasters       IMPORT MFDBType, CopyRasterOpaque ;

VAR d:INTEGER;               (* dummy-Variable *)
    MFDB:MFDBType;           (* Memory-Form-Definition-Block *)

PROCEDURE ObjectEditExtended(Tree:ADDRESS; Object, Typelength:INTEGER;
                             VAR typeline:ARRAY OF CHAR;
                             VAR typestyle:ARRAY OF INTEGER);

VAR vdihandle,linex,liney,curstyle,lasttype,charwidth,charheight:INTEGER;
    finished:BOOLEAN;
    fillat,pxy:PxyArrayType;
    attrib:TextAttrArrayType;

(* ein Rechteck l|schen *)
PROCEDURE ClearRec(x,y,w,h:INTEGER);
BEGIN
  (* in wei~ *)
  d:=SetFillColour(vdihandle,0);
  (* Rechteck definieren *)
  pxy[0]:=x; pxy[1]:=y;
  pxy[2]:=x+w-1; pxy[3]:=y+h-1;
  GrafMouse(MouseOff,NIL);
  (* und zeichnen *)
  DrawBar(vdihandle,pxy);
  GrafMouse(MouseOn,NIL);
  d:=SetFillColour(vdihandle,1);
END ClearRec;

(* ganze Zeile l|schen *)

PROCEDURE ClearLine;
BEGIN
  lasttype:=0;
  FOR d:=0 TO Typelength DO
    typeline[d]:=' ';
    typestyle[d]:=NormalStyle;
  END;
  ClearRec(linex-8,liney,Typelength*charwidth+7,charheight);
END ClearLine;

(* Zeilenteil nach links oder rechts bewegen *)
PROCEDURE MoveLine(Left:BOOLEAN;start,end:INTEGER);
VAR pxy:BigPxyArrayType;
BEGIN
```

Listing 1b

JETZT LIEFERBAR

SALIX PROLOG

FÜR ALLE ATARI ST

mit ausführlichem deutschem Handbuch

Darüberhinaus wurde SALIX PROLOG mit weiteren nützlichen Funktionen versehen. Insgesamt werden ca. 140 eingebaute Funktionen („Primitives") unterstützt. Außerdem wird eine Bibliothek mitgeliefert, die es gestattet, fast alle Funktionen des GEM als PROLOG-Funktionen zu benutzen. SALIX PROLOG hat eine komfortable GEM-Oberfläche mit Drop-Down Menüs und Formularen und einem Window-Management-System. Diese GEM-Oberfläche wurde selbst in SALIX PROLOG geschrieben.

```
Desk  Datei  Edit  Debug  Optionen  Hilfe          158278 Bytes frei
      consult user   consult user
      consult        - Top Level Interpreter        debug & help
      reconsult
      forget                                        fractaltree(N) :-
      dump as ...                                      N>3,
      ----------                                       wind_get(_,10,Wind,_,
      load                                             wind_get(Wind,4,_,_,_
      ----------                                       H1 is 0.1199999999999
      Exit                                             H2 is H/ 4,
                                                       fractaltree(N,L,H2),
                                                       graf_mouse(off),
                                                       fractal([H1|L]),
                                                       graf_mouse(on) .
                                                    fractaltree(1,[],_) :-
                                                       ! .
                                                    fractaltree(N,[H|L],H) :-
                                                       N1 is N-1,
                                                       fractaltree(N1,L,H) .
```

Leistungsdaten:

- *Compiler/Interpreter System.* Der inkrementelle Compiler erzeugt einen Zwischencode, der vom Interpreter abgearbeitet wird.
- **Edinburg Standard**
  Alle im Buch ‚PROLOG — Programming in Logic' von Clocksin & Mellish definierten Funktionen sind verfügbar.
- **zusätzliche eingebaute Funktionen**
  (primitives), insgesamt ca. 140 Funktionen
- **Integer- und Gleitkommaarithmetik**
  mathematische Funktionen, Gleitkomma in doppelter Genauigkeit.
- **Benutzerausgänge:**
  - ***Exception Handler.*** Bei einem Fehler kann der Benutzer mit dem Prädikat 'error' eigene Fehlerroutinen in Kraft setzen.
  - ***Leistungsfähiges Testsystem.*** Das Testsystem ist selbst in PROLOG geschrieben und kann bei Bedarf modifiziert werden.
  - ***portray-Prädikat.*** Das Standardausgabe-Prädikat 'print' kann vom Benutzer modifiziert werden.
  - ***Einbindung benutzergeschriebener Precompiler.*** Die Einbindung erfolgt in die Prädikate 'consult' und 'reconsult'. Beispielsweise können so Bibliotheken, die aus Grammatikregeln oder Regeln für ein Expertensystem bestehen, direkt konsultiert werden.
  - ***Benutzerausgang bei Zeicheneingabe.*** Damit können komfortable Benutzeroberflächen für Expertensysteme angeschlossen werden.
- **Syntax Error Corrector**
  Tritt beim Einlesen eines PROLOG-Ausdrucks (vom Bildschirm oder vom Laufwerk) ein Syntaxfehler auf, erhält der Benutzer Gelegenheit, den Fehler am Bildschirm zu verbessern.
- **Lector.** Ein in PROLOG geschriebenes Hilfsmittel, um PROLOG-Programme auf Tippfehler und falsche Verwendung der eingebauten Funktionen zu prüfen.
- **Anschluß an einen vorhandenen Editor.** (z.B. 1st-Word) Die PROLOG-Sitzung muß nicht verlassen werden, um eine Bibliothek zu editieren.
- **GEM-Library** gestattet die Nutzung fast aller GEM Funktionen von PROLOG aus, damit steht die riesige Grafikbibliothek des GEM in der Form von PROLOG-Prädikaten zur Verfügung.
  Zusätzlich Window Management System mit Pufferung und automatischem Refresh der Fenstertextinhalte.

```
Desk  Edit  File  Options
                 A:\PROLOG.RSC
BUTTON     Desk  Datei  Edit  Debug  Optionen  Hilfe
STRING     Abou  consu  edit  t trac  solveone  gap
EDIT:____        consu  edit  n notr  solveall  dict
EDIT:____  Desk  reconsult            dict ...
           Desk  forget       b back  showterm  listing ...
           Desk  dump as ...  c cree  showvar
           Desk               s skip
           Desk  load         w walk
TEXT       Desk
C                Exit         spy ...
BOXTEXT                       nospy ...
ICON                          debugging
                              nodebug
                              leash
                              ir_mask
```

- **GEM-Top-Level-Interpreter,** komfortable Bedienung des PROLOG-Programmiersystems mit Drop-Down-Menüs und Formularen. Der Gem-Top Level Interpreter ist komplett in PROLOG geschrieben, damit auch für spezielle Anforderungen modifizierbar.
- **Startup-File,** um das PROLOG-System entsprechend den jeweiligen Anforderungen zu konfigurieren.
- Der **volle Adressraum des 68000** wird unterstützt, so volle Ausnutzung des ATARI-Speichers und etwaiger Speichererweiterungen.
- **Automatische Garbage Collection.**
- **Beispiel- und Utility-Bibliotheken.**

DM **198,-**

* unverb. empf. Verkaufspreise

Heim-Verlag
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon: (0 61 51) 5 60 57

Handbuch einzeln 60,— wird beim Kauf angerechnet

WO BEKOMME ICH SALIX PROLOG?

Ein Expertensystem gibt Antwort

Computer: Welches Problem haben Sie?
Leser: Ich suche ein Programm, mit dem ich Probleme der künstlichen Intelligenz lösen kann.
Computer: Wollen Sie auch Expertensysteme, etwa so wie mich programmieren?
Leser: Was sind Expertensysteme?
Computer: Programme wie ich, wo Ihnen der Computer mit Fachwissen ein Problem lösen kann.
Leser: Ja, das ist wohl eine Hauptanwendung.
Computer: Soll dieses Programm auf einem leistungsfähigen modernen 68000 Computer laufen?
Leser: Selbstverständlich, denn die Sprache soll ja schnell sein.
Computer: Soll das Programm auch unter einer benutzerfreundlichen Oberfläche, etwa GEM laufen?
Leser: Das wird wohl optimal sein.
Computer: Wie soll das Preis-Leistungsverhältnis sein?
Leser: Möglichst günstig.
Computer: Da kann ich Ihnen nur das Salix Prolog für den Atari St empfehlen.
Leser: Ist das denn nicht zu teuer?
Computer: Nein, trotz eines guten Leistungsverhältnisses kostet es nur 198,— DM.
Leser: Das ist ja toll. Wo kann ich es bekommen?
Computer: Das ist ganz einfach. Untenstehenden Coupon ausfüllen und an den Heim Verlag, Heidelberger Landstr. 194, 6100 Darmstadt einsenden.

BESTELL-COUPON

Einsenden an: Heim Verlag - 6100 Da.-Eberstadt - Heidelberger Landstr. 194

Bitte senden Sie mir:

___ St. SALIX-PROLOG mit über 250 S. deutschem Handbuch zum Preis von 198,— je Stück.

zzgl. DM 5,— Versandkosten bei NN

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name

Straße

Ort

```
  (* Quell- und Zielbereiche errechnen *)                        ②
  pxy[0]:=linex+start*charwidth;
  pxy[1]:=liney;
  pxy[2]:=pxy[0]+(end-start+1)*charwidth;
  pxy[3]:=pxy[1]+charheight;
  IF Left THEN
    pxy[4]:=pxy[0]-charwidth;
    pxy[6]:=pxy[2]-charwidth;
  ELSE
    pxy[4]:=pxy[0]+charwidth;
    pxy[6]:=pxy[2]+charwidth;
  END;
  pxy[5]:=pxy[1];
  pxy[7]:=pxy[3];
  GrafMouse(MouseOff,NIL);
  (* Bildschirmbereich kopieren *)
  CopyRasterOpaque(vdihandle,3,pxy,ADR(MFDB),ADR(MFDB));
  GrafMouse(MouseOn,NIL);
  (* "|berstehenden" Teil l|schen und Zeileninhalt bewegen *)
  IF Left THEN
    ClearRec(pxy[2],pxy[1],charwidth,charheight);
    IF start=0 THEN
      INC(start);
    END;
    FOR d:=start TO end DO
      typeline[d-1]:=typeline[d];
      typestyle[d-1]:=typestyle[d];
    END;
  ELSE
    ClearRec(pxy[0],pxy[1],charwidth,charheight);
    IF end=Typelength THEN
      DEC(end);
    END;
    FOR d:=end TO start BY -1 DO
      typeline[d+1]:=typeline[d];
      typestyle[d+1]:=typestyle[d];
    END;
  END;
END MoveLine;
(* Cursor bei lasttype zeichnen *)
PROCEDURE SetCursor;
BEGIN
  (* XOR-Modus *)
  d:=SetWritingMode(vdihandle,3);
  (* Cursorbereich errechnen *)
  pxy[0]:=linex+lasttype*charwidth-1;
  pxy[1]:=liney;
  pxy[2]:=pxy[0]+charwidth+1;
  pxy[3]:=pxy[1]+charheight;
  GrafMouse(MouseOff,NIL);
  (* und zeichnen *)
  DrawBar(vdihandle,pxy);
  GrafMouse(MouseOn,NIL);
  (* Replace-Modus *)
  d:=SetWritingMode(vdihandle,1);
END SetCursor;

(* Auf Tasten reagieren *)
PROCEDURE DoTypeWrite;
VAR kret:INTEGER;
    str:ARRAY[0..1] OF CHAR;
BEGIN
  (* auf Eingabe warten *)
  kret:=EventKeyboard();
  (* Sondertasten behandeln *)
  CASE kret DIV 256 OF
        1: (* ESC *)
           SetCursor;
           ClearLine;  (* Zeile l|schen *)
```

```
           SetCursor;                                                 ③
    |  14: (* BS *)
           IF lasttype>0 THEN
             SetCursor;
             MoveLine(TRUE,lasttype,Typelength); (* Bereich verschieben *)
             DEC(lasttype);                      (* und Cursor versetzen *)
             SetCursor;
           END;
    |  28: (* RET *)
           finished:=TRUE;      (* beenden *)
    |  59: (* F1 *)
           IF ODD(curstyle DIV Bold) THEN (* Fettschrift ein-/ausschalten *)
             DEC(curstyle,Bold);
           ELSE
             INC(curstyle,Bold);
           END;
           (* Schriftart setzen *)
           d:=SetGraphicTextEffects(vdihandle,curstyle);
    |  60: (* F2 *)
           IF ODD(curstyle DIV Underline) THEN (* Unterstreichen   *)
             DEC(curstyle,Underline);          (* ein-/ausschalten *)
           ELSE
             INC(curstyle,Underline);
           END;
      GrafMouse(MouseOn,NIL);
      (* Cursor weitersetzen *)
      INC(lasttype);
      SetCursor;
    END;
  END;
END DoTypeWrite;

BEGIN
  (* Men|s und Fensteraktionen ausschalten *)
  WindowUpdate(BeginMouseControl);
  WindowUpdate(BeginUpdate);
  (* VDI-Handle holen *)
  vdihandle:=GrafHandle(charwidth,charheight,d,d);
  (* aktuelle Einstellungen merken *)
  InquireTextAttributes(vdihandle,attrib);
  InquireFillAttributes(vdihandle,fillat);
  (* Objektposition holen *)
  ObjectOffset(Tree,Object,linex,liney);
  (* korrigieren wg. ]berhang nach links bei Kursivschrift ! *)
  INC(linex,6);
  d:=SetWritingMode(vdihandle,1);               (* Replace-Modus *)
  d:=SetGraphicTextColour(vdihandle,1);         (* schwarz       *)
  SetGraphicTextAlignment(vdihandle,0,5,d,d);   (* Bottom-Line   *)
  d:=SetFillColour(vdihandle,1);                (* schwarz       *)
  d:=SetFillInteriorStyle(vdihandle,1);         (* solid         *)
  finished:=FALSE;
  curstyle:=NormalStyle;
  ClearLine;
  SetCursor;
  REPEAT
    DoTypeWrite
  UNTIL finished;
  (* alte Einstellungen wiederherstelen *)
  d:=SetFillInteriorStyle(vdihandle,fillat[0]);
  d:=SetFillColour(vdihandle,fillat[1]);
  d:=SetWritingMode(vdihandle,fillat[3]);
  d:=SetGraphicTextColour(vdihandle,attrib[1]);
  SetGraphicTextAlignment(vdihandle,attrib[3],attrib[4],d,d);
  d:=SetGraphicTextEffects(vdihandle,NormalStyle);
  (* Fensteraktionen und Men|s wieder zulassen *)
  WindowUpdate(EndUpdate);
  WindowUpdate(EndMouseControl);
END ObjectEditExtended;
```

```
BEGIN
  (* MFDB vorbereiten *)
  WITH MFDB DO
    (* Bildschirmadresse holen *)
    pointer:=ScreenPhysicalBase();
    (* Aufl!sung setzen *)
    CASE GetResolution() OF
      Low     : width:=320;
                height:=200;
          (* Schriftart setzen *)
          d:=SetGraphicTextEffects(vdihandle,curstyle);
  | 61: (* F3 *)
          IF ODD(curstyle DIV Italic) THEN (* Kursivschrift ein-/ *)
            DEC(curstyle,Italic);          (* ausschalten         *)
          ELSE
            INC(curstyle,Italic);
          END;
          (* Schriftart setzen *)
          d:=SetGraphicTextEffects(vdihandle,curstyle);
  | 62: (* F4 *)
          IF ODD(curstyle DIV Light) THEN  (* Hellschrift ein-/ *)
            DEC(curstyle,Light);           (* ausschalten       *)
          ELSE
            INC(curstyle,Light);
          END;
          (* Schriftart setzen *)
          d:=SetGraphicTextEffects(vdihandle,curstyle);
  | 75: (* <- *)
          IF lasttype>0 THEN
            SetCursor;
            DEC(lasttype);      (* Cursor nach links *)
            SetCursor;
            curstyle:=typestyle[lasttype];
          END;
  | 77: (* -> *)
          IF lasttype<Typelength THEN
            SetCursor;
            INC(lasttype);      (* Cursor nach rechts *)
            SetCursor;
            curstyle:=typestyle[lasttype];
          END;
  | 83: (* DEL *)
          SetCursor;
          MoveLine(TRUE,lasttype+1,Typelength); (* Bereich verschieben *)
          SetCursor;
  | 114: (* Enter *)
          finished:=TRUE;  (* beenden (Zehnerblock) *)
ELSE
  (* normale Eingabe einf!gen *)
  IF (INTEGER(CHAR(kret))>31) THEN
    SetCursor;
    MoveLine(FALSE,lasttype,Typelength-2); (* Bereich verschieben *)
    typeline[lasttype]:=CHAR(kret);        (* Zeichen und        *)
    typestyle[lasttype]:=curstyle;         (* Stil merken        *)
    IF lasttype=Typelength THEN
      DEC(lasttype)              (* Korrektur am Zeilenende *)
    END;
    (* neues Zeichen ausgeben *)
    d:=SetGraphicTextEffects(vdihandle,curstyle);
    str[0]:=CHAR(kret);
    str[1]:=0C;
    GrafMouse(MouseOff,NIL);
    GraphicText(vdihandle,linex+lasttype*charwidth,liney,str);
```

④

```
                planes:=4;
    | Medium : width:=640;
                height:=200;
                planes:=2;
    | High   : width:=640;
                height:=400;
                planes:=1;
      END;
      (* Bildschirmbreite in Worten errechnen *)
      widthW:=width DIV 16;
      (* Format ist konstant Standard-Format ! *)
      format:=1;
    END;
END AESExtender.
```

⑤

```
MODULE Demo;

FROM SYSTEM IMPORT CODE, ADDRESS;
FROM AESResources IMPORT ResourceObjectFix;
FROM AESForms IMPORT FormCenter;
FROM AESObjects IMPORT ObjectDraw;
FROM AESExtender IMPORT ObjectEditExtended;

(* Durch das Modul Resource steht die ben!tigte Dialogbox im
   Programmcode. Es wurde mit dem RSCMAKER erstellt. Die Baum-
   Adresse steht in diesem Fall in ObjectAddr.
   Es ersetzt die ResourceLoad und ResourceGetAddr Aufrufe bei
   externen RSC Dateien
                                                              *)

MODULE resource;
(* Produced by ResourceMaker 0.10a
   Copyright (c) 1985, 1986 Modula 2 Software Ltd.
   Copyright (c) 1985, 1986 TDI Software Inc.
   Resource file version: 00000H *)

  IMPORT CODE, ADDRESS, ResourceObjectFix;
  EXPORT ObjectAddr;

  CONST
    (* object types that don't have to be relocated *)
    GBOX = 20;
    GPROGDEF = 24;
    GIBOX = 25;
    GBOXCHAR = 27;

  CONST
    (* resource file object counts and indicies *)
    nrObjects = 8;  ixObjects = 134;
    nrTrees = 1;  ixTrees = 326;

  (* resource file data inserted into CODE stream *)
  (*$S-,$P-,$T-*)
  PROCEDURE RESOURCEDATA0;
  BEGIN
    CODE(00000H,00086H,00086H,00086H,00086H,00086H,00024H,00086H);
    CODE(00086H,00146H,00008H,00001H,00000H,00000H,00000H,00000H);
    CODE(00000H,0014AH,04631H,0203DH,02046H,06574H,07400H,04632H);
    CODE(0203DH,02055H,06E74H,06572H,07374H,07269H,06368H,0656EH);
    CODE(00046H,03320H,03D20H,04B75H,07273H,06976H,00046H,03420H);
    CODE(03D20H,04865H,06C6CH,00027H,04F62H,06A65H,06374H,04564H);
    CODE(06974H,04578H,07465H,06E64H,06564H,02700H,04B6FH,06D66H);
    CODE(06F72H,07461H,06265H,06C20H,04564H,06974H,06965H,07265H);
    CODE(06E20H,06D69H,07400H,0FFFFH,00001H,00007H,00014H,00000H);
    CODE(00010H,00002H,01100H,00000H,00000H,00039H,00007H,00002H);
    CODE(0FFFFH,0FFFFH,00014H,00000H,00010H,000FFH,01100H,00002H);
    CODE(00003H,00035H,00001H,00003H,0FFFFH,0FFFFH,0001CH,00000H);
    CODE(00000H,00000H,00024H,00002H,00005H,00009H,00001H,00004H);
    CODE(0FFFFH,0FFFFH,0001CH,00000H,00000H,00000H,0002EH,0000DH);
```

Listing 2

(6)

```
CODE(00005H,00012H,00001H,00005H,0FFFFH,0FFFFH,0001CH,00000H);
CODE(00000H,00000H,00041H,00021H,00005H,0000BH,00001H,00006H);
CODE(0FFFFH,0FFFFH,0001CH,00000H,00000H,00000H,0004DH,0002EH);
CODE(00005H,00009H,00001H,00007H,0FFFFH,0FFFFH,0001CH,00000H);
CODE(00000H,00000H,00057H,00020H,00001H,00014H,00001H,00000H);
CODE(0FFFFH,0FFFFH,0001CH,00020H,00000H,00000H,0006CH,00005H);
CODE(00001H,00019H,00001H,00000H,00086H);
END RESOURCEDATA0;

TYPE OBJECT = RECORD
                next, head: INTEGER;
                tail, type: INTEGER;
                flags, state: INTEGER;
                spec: ADDRESS;
                x, y: INTEGER;
                width, height: INTEGER;
              END;
VAR  i: CARDINAL; o: INTEGER; x: ADDRESS;
     TreeAddr: POINTER TO ARRAY [0..nrTrees-1] OF ADDRESS;
     ObjectAddr: POINTER TO ARRAY [0..nrObjects-1] OF OBJECT;

BEGIN
  (* relocate tree indicies *)
  x := ADDRESS(RESOURCEDATA0) + ixTrees;
  TreeAddr := x;
  FOR i := 0 TO nrTrees-1 DO
    TreeAddr^[i] := TreeAddr^[i] + ADDRESS(RESOURCEDATA0); INC(x,4);
  END;
  (* relocate object specs *)
  x := ADDRESS(RESOURCEDATA0) + ixObjects;
  ObjectAddr := x;
  FOR i := 0 TO nrObjects-1 DO
    WITH ObjectAddr^[i] DO
      IF (type # GBOX)  & (type # GPROGDEF) &
         (type # GIBOX) & (type # GBOXCHAR) THEN
        (* relocate against resource base *)
        spec := spec + ADDRESS(RESOURCEDATA0);
      END;
      (* Fix up the scaling *)
      ResourceObjectFix(ObjectAddr,i) ;  (*0.10a*)
    END;
  END;
END resource;

CONST
  DEMOBOX = 0 ;
     EDIT = 1 ;
editlength = 50;

VAR x,y,w,h:INTEGER;
    line:ARRAY [0..editlength] OF CHAR;
    style:ARRAY[0..editlength] OF INTEGER;

BEGIN
  (* Dialogbox zentrieren *)
  FormCenter(ObjectAddr,x,y,w,h);
  (* Dialogbox zeichnen *)
  ObjectDraw(ObjectAddr,DEMOBOX,5,x,y,w,h);
  (* und Editieren aufrufen *)
  ObjectEditExtended(ObjectAddr,EDIT,50,line,style);
END Demo.
```

Eine Originelle File Selector Box

Nachdem in der ST 3/87 das Programm Tempus vorgestellt wurde, dessen File Selector Box mir sofort gefallen hat, habe ich mich entschlossen, selbst eine ähnlich komfortable zu programmieren.

Jeder kennt die Original Atari File Selector Box. Wer eine Weile mit Programmen wie 1ST WORD unter einer Ram-Disk gerarbeitet hat, kennt das Problem: Man lädt 1ST WORD von Diskette und will einen Text von Ram-Disk laden. Dazu muß man umständlich mit den Cursortasten im Pfadnamen die Diskettenkennung ändern und anschließend mit der Maus das Redraw-Feld anklicken.

Die Lösung:

Diese Box stellt 15 Diskettenstationen zur Auswahl, wobei die angeschlossenen hervorgehoben werden und nur diese auswählbar sind. Nun braucht man nur noch den Knopf der Station anzuwählen, und der Inhalt wird angezeigt. Ebenso verhält es sich mit den Extensions.

Mit dieser Routine können Sie natürlich nicht Ihre gekauften Programme tunen, sondern nur Ihre eigenen benutzerfreundlich machen.

Der Anfang:

Zuerst schnappen Sie sich ein Resource-Construction-Set und bauen die (Dialog-) Box. Diese nennen Sie „SELECT". Mit untenstehender Tabelle und Bild 1 sollte dies eigentlich gelingen. Bitte übernehmen Sie meine Objekt-Namen und die Verwandtschaftsverhältnisse zwischen den einzelnen Objekten. Auch sollten Sie die Stringlängen nicht unterschreiten, sonst gibt's wieder mal Bomben. Die Nummern im .H File spielen dagegen keine Rolle. Jedoch sollten die Disk-Buttons, Extensions und Filenamen richtig geordnet werden. Die Resource speichern Sie unter dem Namen „SELECTOR.RSC" ab.

Die Routine:

Die Routine ist im abgedruckten Programm enthalten. Das Programm

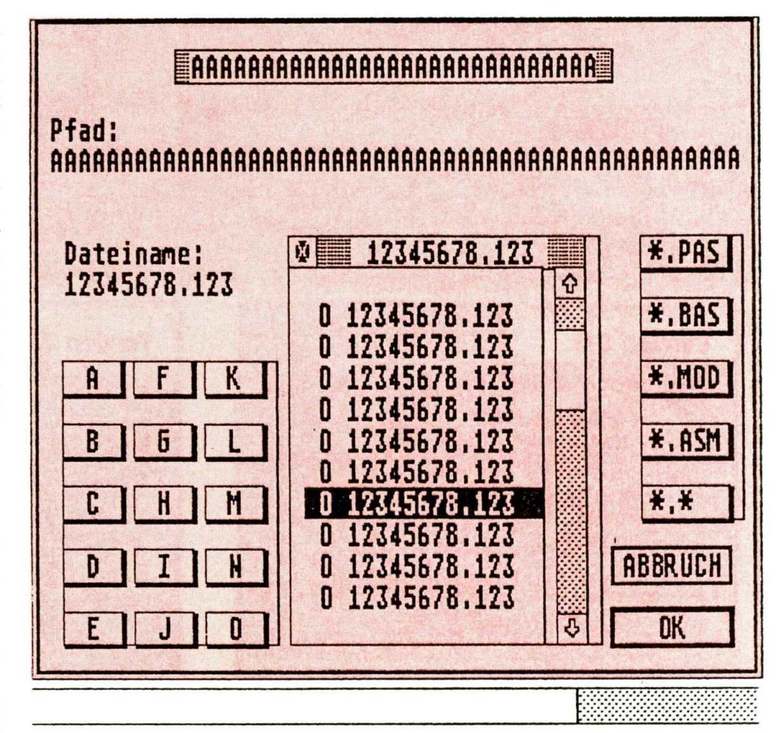


Bild 1

selbst meldet sich in deklaration () beim GEM an, lädt das .RSC File und reserviert Platz für eine Bildschirmseite. Dann wird die File Select Box aufgerufen und die erhaltenen Pfad- und Dateinamen werden ausgedruckt. Schließlich wird das Programm ordnungsgemäß mit der Funktion ende () verlassen.

Der Aufruf für die Box lautet:
fehler =
power_f_sel (pfad,name,info);

Die drei Übergabeparameter sind Pointer auf den Pfadnamen, den Dateinamen und das Selector Info. Sie werden in der Box angezeigt. Nun kann man sich eine Datei auswählen oder andere Directorys anschauen. Verlassen wird die Box mit OK, ABBRUCH oder Doppelklick auf den Dateinamen. Beim Verlassen mit ABBRUCH wird fehler = 1, sowohl pfad als auch name bleiben unverändert. Sonst ist fehler = 0 und die aktuellen Pfad- und Dateinamen werden übermittelt.

Zur Routine:

Wenn Sie den GEM Kurs aufmerksam verfolgt haben, müßten Sie eigentlich durchblicken. Da alle Objekte mit Touchexit ausgestattet sind, wird nach fast jedem Mausblick die Kontrolle wieder von der Routine übernommen. Es werden nun alle Objekte der Reihe nach durchgegangen und geschaut, ob z.B. gerade der Closer ausgewählt wurde. Dementsprechend wird gehandelt und z.B. der Pfadname geändert. Danach wird meistens die Funktion inhalt_anzeigen (pfad,n,anzahl) aufgerufen. Dort wird der Pfad untersucht und n-te Datei als erste angezeigt.

Da ich für die Routine selbst keine globalen Variablen benutzen wollte,

Nummer in .H	Name	Art	Flags	Sonstiges
0	FSTOTAL	Box	Outlined	die äußere Box
1	–	Text	–	Text: „Pfad:“
2	FSPATH	Ftext	Editable	Text: 49 × „A“ (Platzhalter) Template: 49 × „P“
3	–	Text	–	Text „Dateiname:“
4	FSNAME	Fboxtext	Editable	Text: „12345678123“ Template: „pppppppp.ppp“
5	FSDSKBOX	Box	–	Die Box, die die Diskknöpfe umschließt (Bild 1) normal unsichtbar: Border Color 0
6–20	FSAD, FSBD– FSOD	Button	Selectable, Shadowed, Touchexit, Radio Button	Text: „A“, „B“ – „O“ Sind alles Kinder von FSDSKBOX Richtig Ordnen!
21	FSINFO	Boxtext	–	Text: 29 × „A“ Fill setzen (siehe Bild 1) Justification: Center
22	FSTOTBOX	Box	–	Diese Box bildet das „Fenster“. In Bild 1 ist sie rechts etwas größer gezeichnet. 23–39 Kinder von FSTOTBOX
23	FSCLOSER	Boxchar	Touchexit	Char: Der Closer (Ctrl E)
24	FSMOVEDO	Boxchar	Touchexit	Char: Pfeil nach unten (Ctrl B)
25	FSREDRAW	Boxtext	Touchexit	Text: „12345678.123“ Justification: Center Fill gepunktet setzen
26	FSMOVEUP	Bochar	Touchexit	Char: Pfeil nach oben (CTRL A)
27	FSROLL	Box	Touchexit	Fill schwach gepunktet setzen Roll ist die schwach gepunktete Box am rechten „Fensterrand“ Reicht vom oberen zum unteren Pfeil
28	FSSLIDER	Box	Touchexit	Kind von FSROLL Wird vom Programm aus vergrößert ...
29	FSFBOX	Box	–	Beinhaltet die File Namen In Bild 1 rechts etwas kleiner.
30–39	FSFILEO – FSFILE9	Text	Selectable, Radio Button, Touchexit	Text: „0 12345678.123“ Die Leerzeichen sind wichtig. Nach diesen Objekten richten Sie das „Fenster“ aus. 30–39 sind Kinder von FSFBOX
40	FSABBRUC	Button	Selectable, Exit	Text: „ABBRUCH“
41	FSOK	Button	Selectable, Exit, Default	Text: „OK“
42	FSEXTBOX	Box	–	Hat 43–47 als Kinder Unsichtbare Box (Border 0) In Bild 1 sichtbar
43–47	FSEXTEN1 – FSEXTEN5	Button	Selectable, Shadowed, Touchexit, Radio Button	Text: „*.BAS“ oder ähnliches Wichtig: unbedingt 5 Zeichen „*.*“ z.B. wird im Programm auf „*.*“ geändert

Tabelle

wird die Anzahl der Dateien der aktuellen Directorys mit Hilfe von Zeigern übergeben. Alle Doppelklicks werden unterdrückt, bis auf die der Dateiauswahlfelder.

Die Routine inhalt_anzeigen () stellt je nach übergebenen Werten das Inhaltsverzeichnis dar und setzt den Slider richtig.

Wie geht das nun mit dem Inhaltsverzeichnis?

Dazu stellt das GEMDOS zwei Funktionen bereit:

Fsfirst (name,file_attribut)
  gemdos (0×4e)
Fsnext ()
  gemdos (0×4f)
Fgetdta ()
  gemdos (0×2f)

Fsfirst sucht nach einem Eintrag mit dem Namen name. Man kann also auf diese Art und Weise nachschauen, ob sich eine bestimmte Datei auf der Diskette befindet. Ist dies nicht der Fall, so wird eine Fehlernummer (–33) zurückgeliefert. Ist die Datei vorhanden, erhält man die 0 zurück und in einem 44 Byte großen Puffer befinden sich Name, Datum und weitere Informationen. Dieser Puffer hat das Format DTABUF. Die Startadresse wird mit der Funktion Fgetdta () ermittelt. Zu Fsfirst () gehört noch der Parameter file_attribut, der die Art der Datei angibt:

0 les- und beschreibbare Datei
1 schreibgeschützte Datei
2 verborgener Eintrag (im Desktop unterdrückt)
4 verborgener System Eintrag (im Desktop unterdrückt)
8 Volume Label
16 Subdirectory
32 Datei wurde geschrieben und geschlossen

Je nachdem, welche Dateien man angezeigt haben will, addiert man ihre Nummern und erhält so das file_attribut. So liefert z.B. 17 alle Subdirectories, schreibgeschützte und ungeschützte Dateien. Wird nicht der ganze Name angegeben, sondern z.B. „*.C??“, wird die erste der Dateien ermittelt. An die jeweils nächste kommt man mit Fsnext ().

Die Funktion anzahl_files (name) ermittelt so die Anzahl der Einträge, deren file_attribut 17 ist und deren Namen entsprechend dem Ende des Pfadnamen ist.

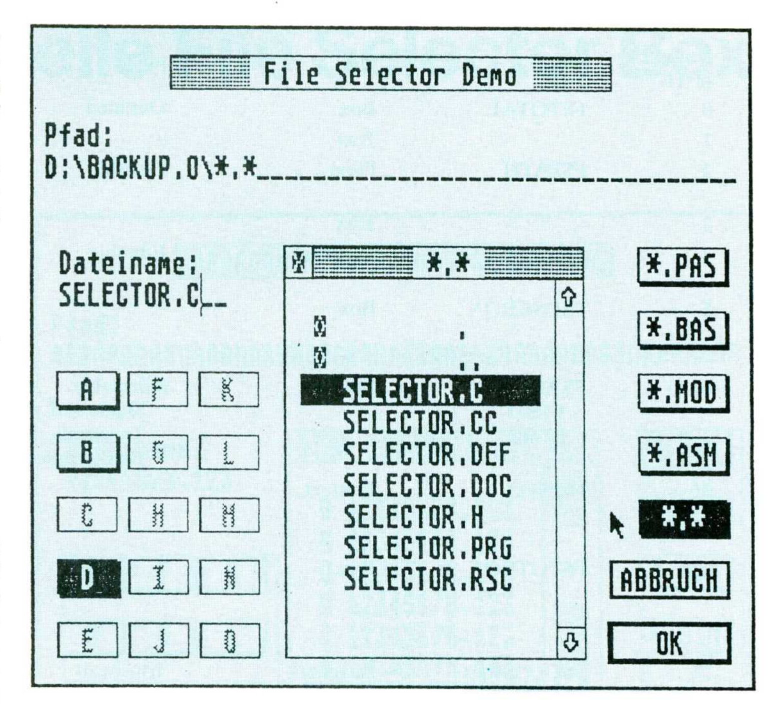


Bild 2:

Falls Sie vom Programm aus die Extension Buttons verändern wollen: in der Funktion power_f_sel () wird dies am Anfang anhand von „*.*“ gezeigt.

Fehler:

Der Unterstrich (_) ... Bitte nicht benutzen, sonst stürzt er ab. Das liegt aber nicht an mir, sonder am GEM. Komisch ist nur, daß er nur abstürzt, wenn die Eingabemaske nicht alle Zeichen zuläßt und sich wie hier auf File-Namen beschränkt (PPP).

Kein Fehler, aber etwas komisch sind in Unterverzeichnissen die ersten beiden Einträge. Das erste ist ein Ordner, der Name besteht aus einem Punkt. Der zweite ist ebenfalls ein Ordner, aber mit zwei Punkten. Im zweiten Eintrag ist das übergeordnete Directory enthalten. Um den Aufwand nicht noch mehr zu vergrößern, werden die beiden einfach mit angezeigt. Man kann sie jedoch nicht auswählen.

Wenn Sie Disketten mit sehr vielen ineinander verschachtelten Subdirectorys haben (Harddisk), sollten Sie darauf achten, daß der Pfadname nicht zu lang wird. Notfalls können Sie die Box breiter machen und mehr Platz für den Pfadnamen reservieren.

Sonstiges:

Im Programm sind noch einige Funktionen enthalten, die auch zur sonstigen GEM Programmierung nützlich sind:

copy (von,nach): Kopiert 32000 Bytes (= 8000 Longwords = 1 Bildschirm) von „von“ nach „nach“. Beide Adressen müssen gerade sein! (Der Bildschirmstart ist sowieso gerade und die Funktion Malloc () zum Anfordern von neuem Speicher liefert auch einen geraden Wert).

dialog_wahl (adresse,min,max): Wird angewendet, um das ausgewählte Element von mehreren Radio Buttons zu ermitteln. Also: die Disk Buttons A–O sind alles Radio Buttons, d.h. es kann nur eines ausgewählt werden. Man übergibt das kleinste (min hier: FSAD) und das größte Element (max hier: FSOD) und erhält das Ausgewählte zurück.

Verwendete Literatur:
ATARI ST GEM von Data Becker
ATARI ST Intern von Data Becker

Ulrich Mast

Ein Bestseller

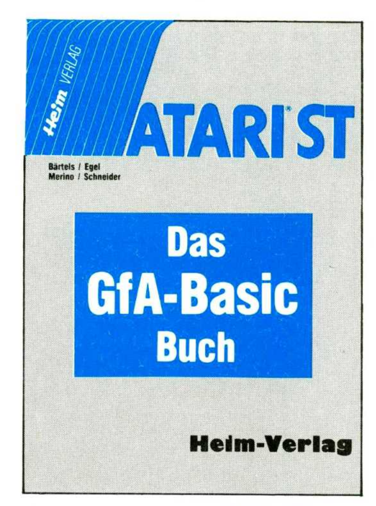
Das GFA-BASIC-Buch

NEU

Die GFA-BASIC-Programmsammlung

Aus dem Inhalt

- Der Umgang mit dem Editor
- Die Vorteile der strukturierten Programmierung ohne Zeilennummern und Pascal-ähnliche Struktur
- Kommunikation mit der Außenwelt (Peripherie)-Floppy, Tastatur-Prozessor, MIDI, der parallele u. serielle Bus, Joystick und Maus
- Mathematik und hohe Genauigkeit
- Schnelle Graphik schnell programmiert – Windows, Sprites, Alertbox, Pull-down Menüs
- Die Systembefehle – einfaches Einbinden von TOS-Befehlen (XBIOS, BIOS, GEMDOS) zur vollständigen Nutzung des Atari-Betriebssystems
- Der Sound – Beschreibung des Soundchips und dessen Programmierung, sogar mit Interrupt-Steuerung



Über 500 Seiten **DM 49,–**

Wichtige Merkmale des Buches

- Die weit über 200 Befehle des neuen GfA-Basic-Interpreters werden mit ihrer Syntax und Bedeutung anhand von vielen Beispielen ausführlich erklärt.
- Das GfA-Basic-Buch ist in Sachgebiete unterteilt, um eine sinnvolle Anwendung der Befehle im Zusammenhang zu ermöglichen. Dabei werden die vielfältigen Fähigkeiten der ATARI ST-Rechner (Graphik, Geschwindigkeit, Window-Technik und Sound) genutzt.
- Das GfA-Basic-Buch ist leicht verständlich geschrieben, so daß auch der Anfänger ohne Probleme seine eigenen Programme erstellen kann.
- Aber auch für den fortgeschrittenen Anwender und für den, der es einmal werden will, bietet das GfA-Basic-Buch die Grundlage zur richtigen Handhabung des Betriebssystems.
- Im Anhang ist eine komplette alphabetische Befehlsübersicht (Nachschlagewerk) enthalten.

★ **Programmdiskette zum Buch: 39,– DM**

---

131 Programme in GFA-BASIC

Folgende Schwerpunkte sind gesetzt:

- Allgemeine Eigenschaften des GFA-Basic
- Zahlen- und Variablenbehandlung
- Dateiverwaltung und Ordner in GFA-Basic
- Grundlegende Befehle
- Schleifentechnik
- Felder (Tabellenverarbeitung u. Matrizen)
- Unterprogramme
- Menü-Steuerung
- Window-Technik
- Zufallszahlen
- Sequentielle Dateien
- RANDOM-Dateien
- Physikalischer Diskettenzugriff
- Textverarbeitung
- Sortierprogramme
- Sortiersystem: SORTMIX
- Fakturiersystem: FAKTURA
- Grafik auf dem Bildschirm u. dem Drucker
- Grafik-Programm: GRAFIKER
- Geometrie, Statistik u. Funktionsplott
- Bildschirmspiel: FLUSS
- Kontoführungsprogramm: KONTO 520
- und viele andere mehr



320 Seiten **DM 49,–**

Wichtige Merkmale des Buches

Anhand von über 130 Programmbeispielen und Erläuterungen lernen und trainieren Sie das Programmieren in GFA-BASIC. Von einfachen, aber grundlegenden Beispielen bis zur ausgereiften Anwendung findet der ST-Besitzer Beispiel-Programme, die den perfekten Einstieg in die Programmiersprache GFA-BSIC leicht und interessant machen.

Alle Beispiele werden so dargestellt, daß das Verständnis für den Programmaufbau gefördert wird, aber auch die Details der GFA-BASIC-Version deutlich werden. Die Programme sind gut dokumentiert, wobei die hervorragenden Möglichkeiten des GFA-BASIC zur strukturierten Programmierung genutzt werden.

★ **Programmdiskette zur Progr.-Sa.: 39,– DM mit 131 Programmen in GFA-BASIC**

Bestellungen bei Ihrem ATARI-Händler (s. Einkaufsführer) oder direkt beim Verlag mit Bestell-Abschnitt

* unverb. empf. Verkaufspreise

Heim-Verlag
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57

Bestell-Abschnitt einsenden an:
HeimVerlag, Heidelberger Landstr. 194, 61 Darmstadt-Eberstadt

Ich bestelle:
- ☐ Das GFA-Basic-Buch .... 49,– DM
- ☐ Programm-Diskette zum Buch .... 39,– DM
- ☐ GFA-BASIC Programmsammlung .... 49,– DM
- ☐ Programm-Diskette zur Progr.-Sammlung 39,– DM

zuzügl. 3,– DM Versandkosten (Scheck oder per Nachnahme)

Name: ____________________

Straße: ____________________

Ort: ____________________

(1)

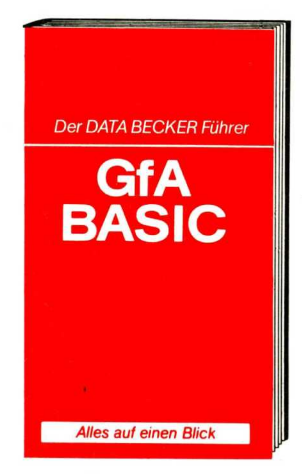
```
 1:  /*******************************************/
 2:  /*                                         */
 3:  /*  FILE-SELECTOR-ROUTINE   für ATARI ST   */
 4:  /*                                         */
 5:  /*   Geschrieben in C (Megamax) von        */
 6:  /*                                         */
 7:  /*   Ulrich Mast                           */
 8:  /*   Wilhelm-Maybach-Str.9                 */
 9:  /*   7303 Neuhausen / Filder               */
10:  /*                                         */
11:  /*******************************************/
12:  #include "gemdefs.h"
13:  #include "osbind.h"
14:  #include "obdefs.h"
15:  #include "selector.h"
16:
17:  #define hide()      graf_mouse(M_OFF,01)
18:  #define show()      graf_mouse(M_ON,01)
19:  #define FEHLER      1
20:  /*******************************************/
21:  typedef struct dta_buffer
22:      {
23:          char    dummy[21];
24:          char    file_attr;
25:          int     time;
26:          int     date;
27:          long    size;
28:          char    name[14];
29:      } DTABUF;
30:
31:  int     contrl[12],intin[128],intout[128];
32:  int     ptsin[128],ptsout[128];
33:  int     work_in[12],work_out[57],handle;
34:  long    fsel_addr;
35:  char    *SCREEN,*SPEICHER,*HILF;
36:  /*******************************************/
37:  main()
38:  {
39:      char pf[80],d[80],was[80];
40:
41:      deklaration();
42:      strcpy(pf ,"A:\\*.*");
43:      strcpy(d  ,"WASWEIS ICH");
44:      strcpy(was," File Selector Demo ");
45:
46:      power_f_sel(pf,d,was);
47:
48:      Cconws(pf);         printf("\n");
49:      Cconws(d);          printf("\n");
50:      Cconin();
51:      ende();
52:  }
53:  /*******************************************/
```

(2)

```
 54:  ende()
 55:  {
 56:      rsrc_free();
 57:      Mfree(SPEICHER);
 58:      v_clsvwk(handle);
 59:      appl_exit();
 60:      exit(0);
 61:  }
 62:  /*******************************************/
 63:  copy(von,nach)
 64:  register long *von,*nach;
 65:  {
 66:      register int laenge;
 67:
 68:      hide();
 69:      laenge=7999;
 70:      asm
 71:      {
 72:          loop:   move.l  (von)+,(nach)+
 73:                  dbf         laenge,loop
 74:      }
 75:      show();
 76:  }
 77:  /*******************************************/
 78:  int
 79:  dialog_wahl(adresse,min,max)
 80:  register    long    adresse;
 81:  register    int         max;
 82:              int         min;
 83:  {
 84:      register    int         i,wahl;
 85:
 86:      for (i=min;i<=max;i++)
 87:          if (select(adresse,i))
 88:              wahl=i;
 89:      return(wahl);
 90:  }
 91:  /*******************************************/
 92:  int
 93:  anzahl_files(exten)
 94:  register    char    *exten;
 95:  {
 96:      register    int    zaehler;
 97:
 98:      if(Fsfirst(exten,17)<0)             return(0);
 99:
100:      for(zaehler=1;Fsnext()>=0;zaehler++);
101:      return(zaehler);
102:  }
103:  /*******************************************/
104:  int
105:  select(tree,which)
106:  register    OBJECT    *tree;
```

Gewußt wo.

Bestimmt kennen Sie das Problem: Sie sind mitten beim Programmieren und suchen eine ganz bestimmte Information. Und dann geht die Sucherei los: Eigentlich kann es nur im Handbuch gestanden haben. Oder war es in einer Zeitschrift? Wenn Sie Informationen gezielt suchen, dann haben wir die richtige Quelle für Sie. Die DATA BECKER Führer. Drei Titel zum ATARI ST sollen Ihnen helfen, Fragen und Probleme schnell zu klären.



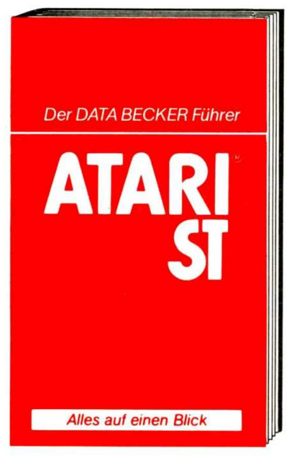
Der DATA BECKER Führer zu GfA-BASIC
254 Seiten, DM 24,80.

Daß das GfA-BASIC eine der leistungsfähigsten BASIC-Versionen ist, die es für den ST gibt, hat sich mittlerweile herumgesprochen. Ein mächtiger Befehlssatz sorgt dafür, daß auch in einer Sprache wie BASIC fast kein Problem mehr ungelöst bleibt. Bis auf das eine: sich die genaue Syntax jedes Befehls zu merken. Aber dafür gibt es ja den DATA BECKER Führer zum GfA-BASIC. Eine komplette Befehlsübersicht inklusive aller nötigen Parameter und Syntax-Erläuterungen sorgt dafür, daß Sie jeden Syntax Error sofort überprüfen können. Das gilt natürlich auch für die 34 neuen Befehle der GfA-BASIC Version 2.0! So haben Sie die Chance, sich voll und ganz auf das Wichtigste zu konzentrieren: auf Ihr Programm.



Der DATA BECKER Führer zu 1st Word
196 Seiten, DM 24,80.

Arbeiten Sie mit einem 1st Programm? Mit 1st Word, 1st Word Plus, 1st Lektor, 1st Mailmaster und/oder 1st Spooler? Dann sollten Sie auf diesen DATA BECKER Führer nicht verzichten. Denn hier finden Sie die Antworten zu allen brennenden Fragen: Wie installiert man einen Drucker? Lassen sich Grafiken einbinden? Wie lassen sich die Programme untereinander verknüpfen? Gibt es Schnittstellen zu anderen Programmen? Egal worüber Sie mehr wissen wollen: Ein Blick in diesen DATA BECKER Führer genügt. Dabei ist es gleich, ob Sie mit der englischen oder deutschen Version arbeiten: dieser DATA BECKER Führer erklärt beide Versionen.



Der DATA BECKER Führer zum ATARI ST
240 Seiten, DM 29,80.

Vor jedem ATARI-Programmierer liegt eine fast unüberschaubare Flut von Routinen, die er nutzen kann: VDI, GEMDOS, BIOS, XBIOS oder AES. Der DATA BECKER Führer zum ATARI ST sorgt dafür, daß Sie den Überblick nicht verlieren. Das gilt natürlich nicht nur für die Systembibliotheken: So finden sich zum Beispiel auch die Befehle und Funktionen des ST BASIC oder eine Liste der Fehlermeldungen. Der DATA BECKER Führer zum ATARI ST ist ein unentbehrliches Nachschlagewerk für all die langen Nächte vor dem ST, in denen man sonst niemanden mehr erreichen kann.

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1

Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name

Straße

Ort

ST 67

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

```
107:  register    int      which;
108:  {                                                                  ③
109:      return(((tree+which)->ob_state&SELECTED)?1:0);
110:  }
111:  /******************************************************************/
112:  deklaration()
113:  {
114:      int i;
115:
116:      appl_init();                                  /* gem anmelden */
117:      for(i=0;i<10 ;work_in[i++]=1 );
118:      work_in[10]=2;
119:      v_opnvwk(work_in,&handle,work_out);
120:      graf_mouse(ARROW,0l);
121:      if(!rsrc_load("SELECTOR.RSC"))                 /* RSC laden */
122:      {
123:          form_alert(1,"[1][Fatal Error ... |RSC-File fehlt ... ][Sorry]");
124:          ende();
125:      }
126:      if(!rsrc_gaddr(0,SELECT    ,&fsel_addr ))
127:      {
128:          form_alert(1,"[1][Irgend etwas im|RSC-File stimmt nicht.][Sorry]");
129:          ende();
130:      }
131:      SPEICHER=(char*)Malloc(33000l);
132:      if(!SPEICHER)
133:      {
134:          form_alert(1,"[1][Nicht genug Speicher ...][Sorry]");
135:          ende();
136:      }
137:      HILF=SPEICHER;                      /* hilfsbildschirmspeicher */
138:      SCREEN=(char*) Logbase();           /* bildschirmstart         */
139:  }
140:  /******************************************************************/
141:  power_f_sel(pfad,name,info)
142:  char    *pfad,*name,*info;
143:  {
144:      int     x,y,w,h,abbruch,ab,wahl,i,k,l,disk,ab_dat,max_anz,pos;
145:      char    *text,*string,dsk,hilf[80];
146:      OBJECT  *tree;
147:      TEDINFO *ted;
148:
149:      tree=(OBJECT  *) fsel_addr;
150:
151:      ted=(TEDINFO *) ((tree+FSINFO)->ob_spec);
152:      strcpy((ted)->te_ptext,info);
153:      ted=(TEDINFO *) ((tree+FSNAME)->ob_spec);
154:      strcpy((ted)->te_ptext,name);
155:      ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
156:      strcpy((ted)->te_ptext,pfad);
157:
158:      ted=(TEDINFO *) ((tree+FSREDRAW)->ob_spec);
159:      strcpy((ted)->te_ptext," ");
```

```
160:      strcpy((tree+FSEXTEN5)->ob_spec,"*.*");
161:                                                                     ④
162:      k=Dsetdrv(Dgetdrv());                        /* angeschlossene   */
163:      for(l=1,disk=FSAD;disk<=FSOD;l*=2,disk++)    /* drives           */
164:      {                                            /* anzeigen         */
165:          if(k & l)
166:              objc_change(fsel_addr,disk,0,x,y,w,h,SHADOWED,0);
167:          else
168:              objc_change(fsel_addr,disk,0,x,y,w,h,DISABLED,0);
169:      }
170:      dsk=*pfad;
171:      disk=(int)(dsk-'A');
172:      if((disk<0)||(disk>15))              return(FEHLER);
173:      objc_change(fsel_addr,disk+FSAD,0,x,y,w,h,SHADOWED+SELECTED,0);
174:      Dsetdrv(disk);
175:
176:      ab_dat=1;                                  /* erste datei anzeigen */
177:      for(i=FSFILE0;i<=FSFILE9;i++)              /* dateiennamen löschen */
178:      {
179:          ted=(TEDINFO *) ((tree+i)->ob_spec);
180:          text=(ted)->te_ptext;
181:          *text=0;
182:      }
183:      copy(SCREEN,HILF);                         /* bildschirm retten    */
184:      form_center(fsel_addr,&x,&y,&w,&h);
185:      form_dial(0,319,199,2,2,x,y,w,h);
186:      form_dial(1,319,199,2,2,x,y,w,h);
187:      objc_draw(fsel_addr,0,3,x,y,w,h);
188:      inhalt_anzeigen(pfad,ab_dat,&max_anz);
189:      while(1)
190:      {
191:          abbruch=form_do(fsel_addr,FSNAME);
192:          ab=abbruch & 0x7fff;                   /* ohne doppelklick     */
193:
194:          if((ab==FSOK)||(ab==FSABBRUC))         /* exit button          */
195:          {
196:              objc_change(fsel_addr,ab,0,x,y,w,h,0,0);
197:              break;
198:          }
199:          if((ab>=FSAD)&&(ab<=FSOD))             /* disk button          */
200:          {
201:              wahl=dialog_wahl(fsel_addr,FSAD,FSOD);
202:              disk=(wahl-FSAD);
203:              if((disk<0)||(disk>15))         return(FEHLER);
204:
205:              dsk=(char)(disk+'A');
206:              ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
207:              *(ted->te_ptext)=dsk;                   /* pfad ändern      */
208:              Dsetdrv(disk);                          /* drive setzen     */
209:              objc_draw(fsel_addr,FSPATH,0,x,y,w,h);  /* pfad redraw      */
210:              inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
211:              abbruch=ab;                             /* kein doppelklick */
212:          }
```

```
213:        if((ab>=FSEXTEN1)&&(ab<=FSEXTEN5))             /* entension button */
214:        {                                                                  (5)
215:            wahl=dialog_wahl(fsel_addr,FSEXTEN1,FSEXTEN5);
216:            ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
217:            i=strlen(ted->te_ptext);
218:            for(string=(ted->te_ptext)+i;string>=(ted->te_ptext);string--)
219:                if(*string=='\\')                  break;
220:            string++;
221:            strcpy(string,(tree+wahl)->ob_spec);
222:            objc_draw(fsel_addr,FSPATH,0,x,y,w,h);
223:            inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
224:            abbruch=ab;
225:        }
226:        if((ab>=FSFILE0)&&(ab<=FSFILE9))              /* dateiname        */
227:        {
228:            wahl=dialog_wahl(fsel_addr,FSFILE0,FSFILE9);
229:            ted=(TEDINFO *) ((tree+wahl)->ob_spec);
230:            string=ted->te_ptext;
231:            if(*string!=0)                          /* eintrag vorhanden */
232:            {
233:                if(*(string+1)==7)                  /* ordner            */
234:                {
235:                    ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
236:                    i=strlen(ted->te_ptext);
237:                    for(string=(ted->te_ptext)+i;string>=(ted->te_ptext);string--)
238:                        if(*string=='\\')          break;
239:                    strcpy(hilf,string);            /* extension retten */
240:                    string++;
241:
242:                    ted=(TEDINFO *) ((tree+wahl)->ob_spec);
243:                    text=ted->te_ptext;
244:                    text+=3;
245:                    for(i=0;i<12;i++)
246:                    {
247:                        if(*text==0)
248:                            break;
249:                        if(*text!=' ')
250:                            *string++=*text;
251:                        text++;
252:                    }
253:                    strcpy(string,hilf);
254:                    objc_draw(fsel_addr,FSPATH,0,x,y,w,h);
255:                    ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
256:                    inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
257:                    abbruch=ab;                     /* kein doppelklick */
258:                }
259:                else                                /* datei            */
260:                {
261:                    string+=3;
262:                    ted=(TEDINFO *) ((tree+FSNAME)->ob_spec);
263:                    text=ted->te_ptext;
264:                    for(i=0;i<8;i++)
265:                        *text++=*string++;
```

```
266:                    string++;
267:                    for(i=0;i<3;i++)
268:                        *text++=*string++;                                 (6)
269:                    *text=0;
270:                    objc_draw(fsel_addr,FSNAME,0,x,y,w,h);
271:                }
272:            }
273:        }
274:        if(ab==FSREDRAW)                                /* redraw         */
275:        {
276:            ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
277:            inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
278:            abbruch=ab;
279:        }
280:        if(ab==FSMOVEUP)                                /* pfeil nach oben */
281:        {
282:            if(ab_dat>1)
283:            {
284:                ab_dat--;
285:                ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
286:                inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
287:            }
288:            abbruch=ab;                                 /* kein doppelklick */
289:        }
290:        if(ab==FSMOVEDO)                                /* pfeil nach unten */
291:        {
292:            if(ab_dat<(max_anz-9))
293:            {
294:                ab_dat++;
295:                ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
296:                inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
297:            }
298:            abbruch=ab;                                 /* kein doppelklick */
299:        }
300:        if(ab==FSSLIDER)                                /* slider         */
301:        {
302:            if(max_anz>10)
303:            {
304:                pos=graf_slidebox(tree,FSROLL,FSSLIDER,1)+500/(max_anz-10);
305:                ab_dat=((max_anz-10)*pos)/1000;
306:                ab_dat++;
307:
308:                ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
309:                inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
310:            }
311:            abbruch=ab;                                 /* kein doppelklick */
312:        }
313:        if(ab==FSROLL)                                  /* roll - balken   */
314:        {
315:            graf_mkstate(&i,&pos,&i,&i);            /* nur y-pos wichtig */
316:            objc_offset(tree,FSSLIDER,&i,&k);
317:            if(pos>k)
318:            {
```

(7)

```
319:                ab_dat+=10;
320:                if(ab_dat>(max_anz-9))      ab_dat=max_anz-9;
321:            }
322:            else
323:            {
324:                ab_dat-=10;
325:                if(ab_dat<1)                ab_dat=1;
326:            }
327:            ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
328:            inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
329:            abbruch=ab;                         /* kein doppelklick */
330:        }
331:        if(ab==FSCLOSER)                        /* schließfeld     */
332:        {
333:            ted=(TEDINFO *) ((tree+FSPATH)->ob_spec);
334:            i=strlen(ted->te_ptext);
335:            for(string=(ted->te_ptext)+i;string>=(ted->te_ptext);string--)
336:                if(*string=='\\')               break;
337:            for(text=string-1;text>=(ted->te_ptext);text--)
338:                if(*text=='\\')                 break;
339:            if(*text=='\\')
340:            {
341:                string++;
342:                text++;
343:                strcpy(text,string);
344:                objc_draw(fsel_addr,FSPATH,0,x,y,w,h);
345:            }
346:            inhalt_anzeigen(ted->te_ptext,ab_dat,&max_anz);
385:    DTABUF  *puffer;
386:
387:    tree=   (OBJECT *)      fsel_addr;
388:    puffer= (DTABUF *)      Fgetdta();
389:
390:    i=strlen(pfad);
391:    for(string=pfad+i;string>=pfad;string--)
392:        if(*string=='\\')       break;
393:    name=string+1;
394:    for(string=pfad+2,i=0;string<name;string++,i++)
395:        hilf[i]=*string;
396:    hilf[i]=0;
397:    Dsetpath(hilf);
398:
399:    *anzahl=anzahl_files(name);
400:    if(*anzahl<=10)
401:    {
402:        ((tree+FSSLIDER)->ob_y)     =0;
403:        ((tree+FSSLIDER)->ob_height)=((tree+FSROLL)->ob_height);
404:        abnr=1;
405:    }
406:    else
407:    {
408:      (tree+FSSLIDER)->ob_y=((tree+FSROLL)->ob_height*(abnr-1))/(*anzahl);
409:      (tree+FSSLIDER)->ob_height=((tree+FSROLL)->ob_height*10)/(*anzahl);
```

(8)

```
410:        }
411:        ted=(TEDINFO *) ((tree+FSREDRAW)->ob_spec);
412:        text=(ted)->te_ptext;
413:        strcpy(text," ");                              /* sieht besser aus  */
414:        strcat(text,name);
415:        strcat(text," ");
416:
417:        fehler=Fsfirst(name,17);
418:        for(i=2;i<abnr;i++)
419:            if(fehler>=0)
420:                fehler=Fsnext();
421:        for(i=FSFILE0;i<=FSFILE9;i++,abnr++)
422:        {
423:            if((abnr!=1)&&(*anzahl>0))
424:                fehler=Fsnext();
425:            ted=(TEDINFO *) ((tree+i)->ob_spec);
426:            text=(ted)->te_ptext;
427:            if((fehler>=0)&&(*anzahl>0))
428:            {
429:                (tree+i)->ob_flags |=SELECTABLE;        /* auswählbar     */
430:                if(puffer->name[0]==46)
431:                    (tree+i)->ob_flags &= ~TOUCHEXIT;   /* kein touchexit */
432:                else
433:                    (tree+i)->ob_flags |= TOUCHEXIT;    /* touchexit      */
434:
435:                *text++=32;
436:                *text++=((puffer->file_attr==16) ? 7 : 32);
437:                *text++=32;
438:                for(k=0,l=0;(puffer->name[k]!='.')&&(l<8);l++)
439:                    *text++=puffer->name[k++];
440:                for(;l<8;l++)
441:                    *text++=32;
442:                for(l=0;l<4;l++)
443:                    *text++=puffer->name[k++];
444:                *text=0;
445:            }
446:            else
447:            {
448:                (tree+i)->ob_flags &= ~SELECTABLE;  /* nicht auswählbar */
449:                (tree+i)->ob_flags &= ~TOUCHEXIT;   /* kein touchexit   */
450:                (tree+i)->ob_state  = NORMAL;       /* nicht ausgewählt */
451:                *text=0;
452:            }
453:        }
454:        objc_draw(fsel_addr,FSTOTBOX,2,1,1,500,380);  /* filebox redraw  */
455:    }
456:    /*****************************   ENDE   ******************************/
```





SHUTTLE II

Wollten Sie nicht schon immer der Kommandant eines SPACE SHUTTLES sein? - jetzt haben Sie die Gelegenheit dazu. Ein defekter Kommunikations-Satellit soll aus dem Weltraum geborgen werden, doch bis dorthin ist es ein weiter Weg. Ihre Aufgaben und Einsatzgebiete sind verschiedenster Natur. Als erstes muß der Abflugpunkt bestimmt werden. Dabei werden ebenfalls die Wetterbedingungen (Wolkenbildung, Windstärke, Richtung) für den Start eingestellt. Je höher die Werte gewählt werden, desto schwieriger ist der Start, aber dann gibt es auch die meisten Punkte. Nachdem die Startphase durchgeführt ist, wird die Landung auf die gleiche Weise geplant.

Nun beginnt der eigentliche Start, bei dem Sie die Rolle des Piloten übernehmen. Sie sitzen in Ihrer Kabine und schauen aus dem Fenster auf die Umgebung des Startplatzes. Der Countdown beginnt. Nun ist eine Flugbahn, die auf einem kleinen Monitor erscheint, möglichst genau nachzufliegen. Mittels drei Reglern steuern Sie diesen Vorgang. Jede Abweichung kostet Punkte.

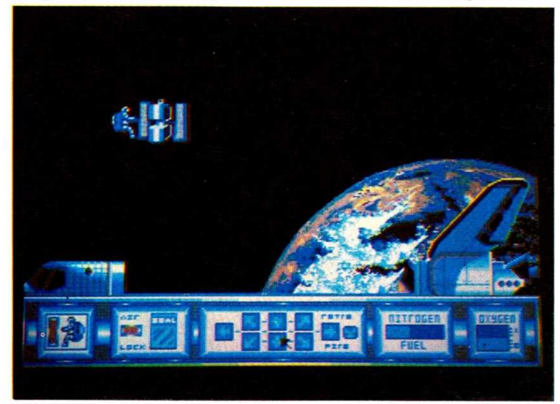
Während des Fluges müssen zu einem bestimmten Zeitpunkt die Triebwerke und später auch der Außentank abgeworfen werden. Wenn diese Aktionen nicht von Hand ausgeführt werden, dann erledigt es der Bord-Computer, allerdings erhält man dann keine Punkte.

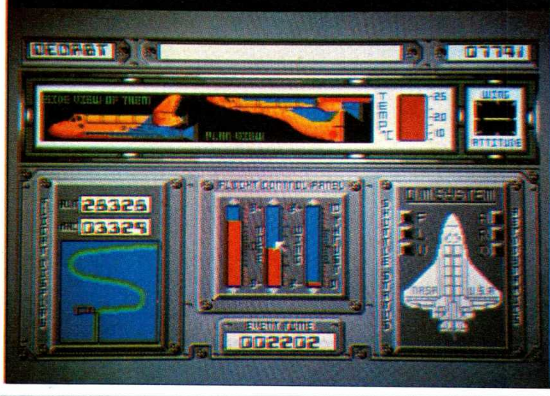
Wenn der Endpunkt des Fluges erreicht ist, muß der genaue Ort des Satelliten ermittelt werden. Dazu wird ein spezielles Führungssystem eingeschaltet, das die Lage des Objektes in allen drei Achsen anzeigt, sofern es nicht außerhalb des Bereiches liegt. Wenn alle drei Werte eingeregelt sind, sind Sie nahe genug am Satelliten, um ihn mit ihrem Jet-Pack erreichen zu können. Dies ist dann auch die nächste Phase Ihres Unternehmens. Nachdem die Jet-Packs mit Treibstoff und Sauerstoff gefüllt sind, werden die Frachtraumtüren geöffnet und Sie können ins Weltall schweben. In einiger Entfernung schwebt der defekte Satellit herum, der nun 'eingefangen ' werden muß. Danach wird er in den Frachtraum gebracht.

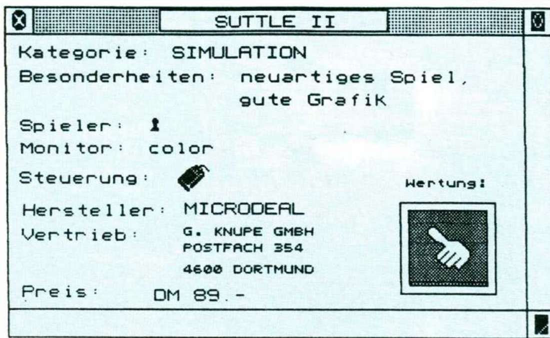
Die Mission ist jetzt aber noch nicht beendet, denn der schwierigste Teil kommt noch - der Rückflug.

Um beim Eintritt in die Atmosphäre nicht zu verglühen, muß das Raumschiff in einem S-turn abgebremst werden. Die Steuerung ist dabei sehr schwierig. Auf einem speziellen Monitor können die verschiedenen Temperaturzonen des Shuttles kontrolliert werden. Wenn dieses Unternehmen geglückt ist, muß das Shuttle nur noch gelandet werden. Also Landebahn anpeilen, Fahrwerk ausfahren und runter.

SHUTTLE II ist ein interessantes Unternehmen, in dem viele verschiedene Szenen durchgespielt werden müssen. Jede Szene hat dabei ihre Eigenheiten und Kniffe, so daß nicht so schnell Langeweile aufkommen kann.







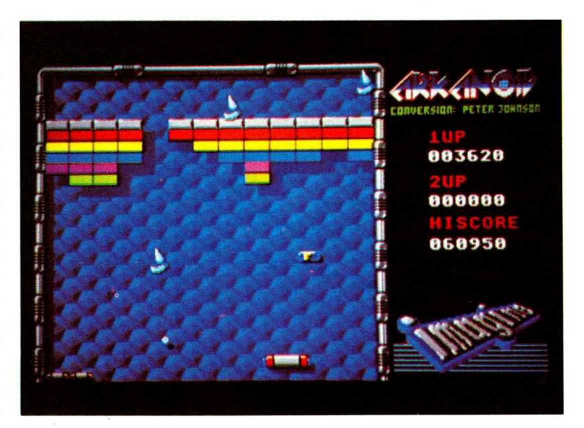
SUTTLE II

Kategorie: SIMULATION
Besonderheiten: neuartiges Spiel, gute Grafik
Spieler: 1
Monitor: color
Steuerung:
Hersteller: MICRODEAL
Vertrieb: G. KNUPE GMBH, POSTFACH 354, 4600 DORTMUND
Preis: DM 89.-
Wertung:

ARKANOID

Der Name wird nur Eingeweihten etwas sagen, denn ARKANOID ist die Umsetzung eines beliebten Spielhallenautomaten. Die Grundidee des Spiels geht geht auf 'Blockbuster' oder auch 'Break Out' zurück, allerdings hat hier eine enorme Wandlung stattgefunden. Wie schon in der Urfassung wird der 'Energieball' immer noch mit einem horizontal bewegbaren Schläger abgelenkt. Er trifft dann auf einen Stein der Mauer und löst diesen auf. Durch ständiges Ablenken des Balls in die verschiedenen Richtungen wird somit die Mauer abgebaut, und der nächste Level ist erreicht.

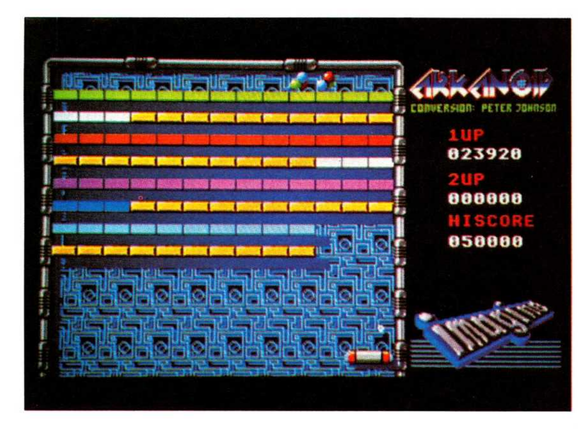
Soweit - so gut, doch das ist natürlich nicht alles, denn die Autoren von ARKANOID haben sich einige spannende Features einfallen lassen, die dieses Spiel gewaltig aufputschen. Dazu gehören erst einmal die Energiekapseln, die sich beim Zerstören der Mauer lösen können. Bei Berührung mit dem Schläger verleihen Sie diesem sehr hilfreiche Fähigkeiten. Als 'Catcher' können die Bälle festgehalten werden, bis ein geeigneter Abschußplatz gefunden ist. Es kann aber auch passieren, daß plötzlich drei Energiekugeln in der Gegend herumsausen, dann wird es ganz schön unübersichtlich. Aber erst wenn die rote Laserkapsel berührt wird, geht es richtig rund, denn jetzt können die Bausteine auch abgeschossen werden. In der Hitze des Gefechts darf man jedoch die Energiekugel nicht vergessen, die ständig die Aufmerksamkeit des Spielers fordert und nur kurzzeitig abzulenken ist. Die 'Big'-Kapsel vergrößert den Schläger und ist somit ein geeignetes Mittel, um sicher jede Figur zu knacken. Die letzte Kapsel ist diejenige, die auch nur sehr selten auftaucht. Mit ihrer Hilfe kann die Wand zum nächsten Spiellevel sofort durchbrochen werden.



Bis jetzt sind nur Objekte erwähnt worden, die das Spiel erleichtern, es gibt jedoch auch solche, die genau das Gegenteil können. Diese Objekte haben die verschiedensten Formen und bewegen sich unregelmäßig über den Bildschirm, wenn sie den Energieball erwischen, lenken sie ihn heimtückisch ab. Dies führt dazu, daß man mit dem Schläger ins Leere läuft und der Energieball in ihn einschlägt. Die verschiedenen Bilder bzw. Level sind sehr abwechslungsreich und zum Teil auch sehr schwer. Schon im ersten Level gibt es Steine, die zweimal getroffen werden müssen, bevor sie verschwinden. Später kommen noch andere hinzu, die sich gar nicht mehr wegschießen lassen. Die Energiekugel muß dann in verschlungenen Bahnen gelenkt werden, deren Eingänge nur schwer zu treffen sind. Mit der Zeit 'erarbeitet' man sich jedoch so manchen Kniff, der diese Aufgaben leichter macht.

Die Umsetzung vom Spielhalleautomaten auf den ST ist bei ARKANOID sehr gut gelungen. Dies betrifft den Abbau der verschiedenen Level und auch die Steuerung des Schlägers. Selbst das monotone 'Bing' der Kugel wurde nicht vergessen, so daß man direkt umsteigen kann.

ARKANOID ist ein unheimlich fesselndes Spiel, denn seitdem es in der Redaktion eingetroffen ist, ist ein ST-Rechner unentwegt blockiert. Ständig versuchen wir Spielsüchtigen, die HiScore-Tabelle zu erobern und die Rivalen auf die hinteren Plätze zu verbannen. Sollten Sie also nur einen Computer haben, dann sind Sie jetzt vor Ihren spielsüchtigen Freunden gewarnt, die sicherlich sofort auftauchen werden, wenn Sie das Spiel haben, und die dann nur noch schwer wieder loszuwerden sind.



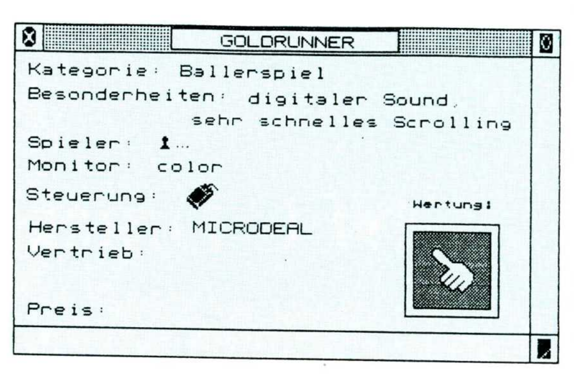




ARKANOID

Kategorie: Geschicklichkeitsspiel
Besonderheiten: sehr spannendes Spiel
mit vielen Optionen
akzeptabler Preis
Spieler: 1
Monitor: color
Steuerung:
Wertung:
Hersteller: IMAGINE
Vertrieb: G. KNUPE GMBH
POSTFACH 354
4600 DORTMUND
Preis: DM 49.50.-

GOLDRUNNER



GOLDRUNNER ist ein superschnelles Ballerspiel von MICRODEAL. Ein sehr wendiges Raumschiff überfliegt die Ringwelten der Tritonen, die unbedingt vernichtet werden müssen. Natürlich sind Sie der Einzige, der dies schaffen könnte, und deshalb geht es auch gleich los. Beim Überfliegen eines Ringes muß alles zerstört werden, was nur geht. Deshalb wird fast ständig aus vollen Rohren geschossen. Mit wahnsinnig hoher Geschwindigkeit wird die Gegend überflogen, doch ist dabei die Gefahr sehr groß, an einem der hohen Gebäude zu zerschellen. Aber es gibt noch andere Gefahren: urplötzlich tauchen feindliche Raumschiffverbände auf, die zwar nicht auf Rammkurs sind, dafür aber heimtückische Raketen abschießen. Wenn mehrere Treffer eingesteckt worden sind, verliert man entweder die Laserkanonen oder den Beschleunigungs-Booster. Dann ist bald der Zeitpunkt erreicht, an dem nichts mehr funktioniert und das Schiff letztlich explodiert.

Das Spiel ist grafisch gut gemacht, und auch die Ohren kommen nicht zu kurz. Die Begleitmusik ist fetzig und wird durch recht verständliche Spracheinlagen effektvoll ergänzt. GOLDRUNNER fasziniert durch sein schnelles, ruckfreies Scrolling und die eleganten Wendemanöver des Raumschiffs. Ansonsten ist es aber ein eher einfaches Ballerspiel mit wenig spannenden Momenten.

WERNER MACH HIN





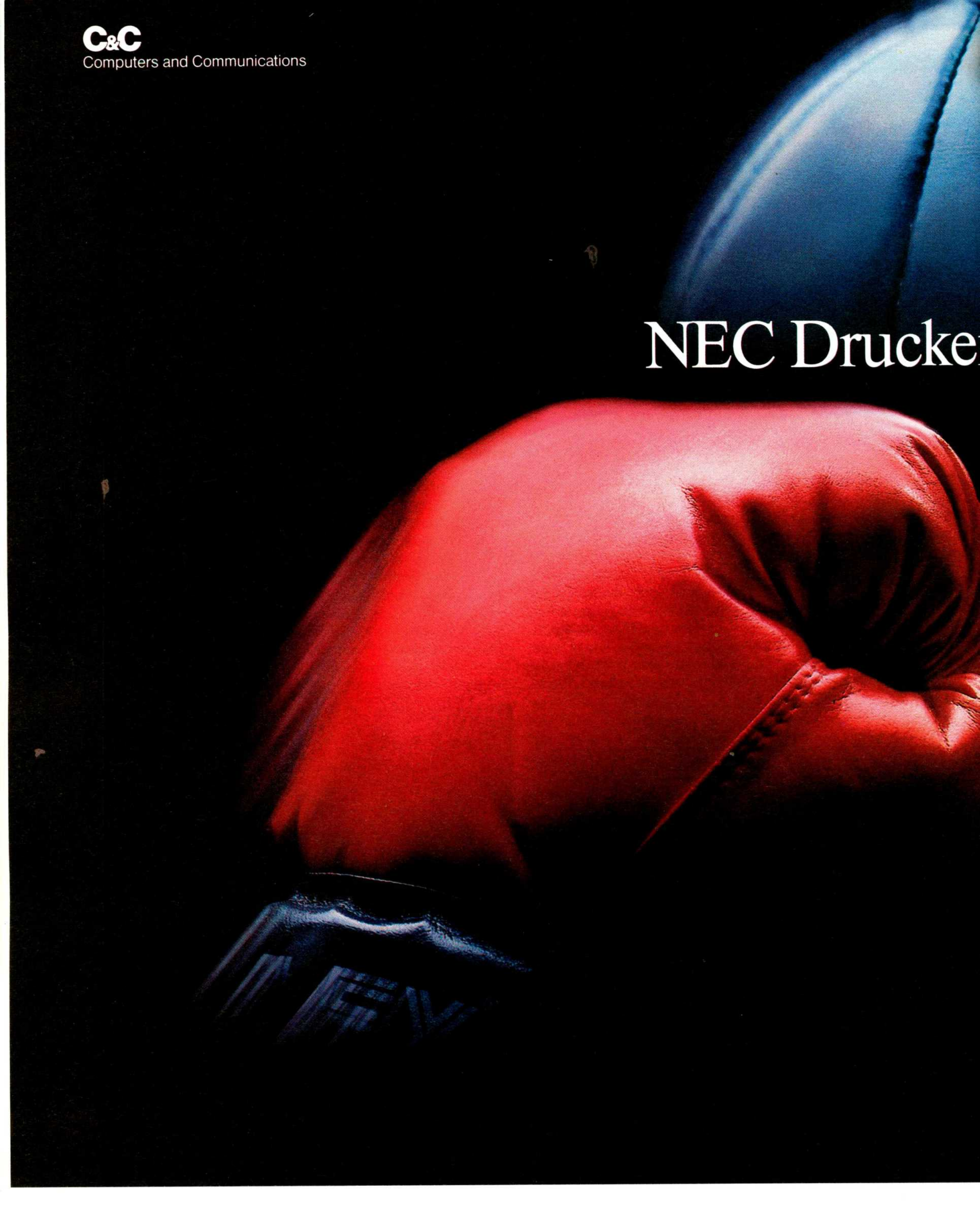
Schon lange Zeit ist es angekündigt worden - jetzt endlich ist es da - das Computerspiel zu WERNER, dem Comic-Helden.

Erste Disziplin: Meiern mit WERNER, Meiern ist ein Würfelspiel - dem 'schummeln' bzw. dem 'Mäxchen' nicht unähnlich. Allerdings braucht man für das Spiel einen eigenen Würfelbecher. Die gewürfelte Zahl teilt man WERNER mit. Dieser ist wirklich ein verwegener Stratege, denn er hält fast alle Vorgaben. Andererseits ist er auch ein großartiger Schummler, der nur knurrend seine Niederlagen eingesteht.

Als nächstes stehen zwei Auto-Touren an, bei denen man sich vor allerlei herumfliegendem Zeug, wie Zündschlüsselabziehern, frei herumfliegenden Bohrmaschinen und Melonen in acht nehmen muß, wenn man die Fahrt fortsetzen will. Ähnlich ist auch die Panic-Tour, wo jedoch bestimmte Gegenstände aufgesammelt werden können, die bei Wagenschäden helfen. Bei beiden Touren geht es außerdem noch darum, möglichst viele Flaschen des begehrten Flens auf eine Fete mitzubringen. Allerdings ist WERNER auch jederzeit dazu bereit, eine Unmenge des begehrten Gerstensaftes sofort auszutrinken, wie wir bei der nächsten Station des Spiels sehen werden.

'Schüssel bauen' ist sicherlich eines der typischsten Beschäftigungen von WERNER. Aus einer Unmenge von Bauteilen kann sich WERNER seine Lieblingsmaschine zusammenbauen. Mit Sicherheit wird das die viermotorige HOREX sein, die als 'RED PORSCHE KILLER' bereits legendär geworden ist. Die Aufgabe ist allerdings am Anfang recht schwierig, weil man nicht weiß, mit welchen Teilen man beginnen soll. Mit der Zeit kommt man allerdings dahinter, und das Ganze wird zum Kinderspiel. Letzte Disziplin des 'bröseligen Fünfkampfes' sind die Nebelfahrt, bei der alles wirklich schleierhaft ist. WERNER hockt auf seinem Bock und versucht, den entgegenkommenden Fahrzeugen auszuweichen, allerdings ist die Zahl der Geisterfahrer heutzutage sehr groß.

Im Gegensatz zum Comic, der wirklich bärenstark ist, ist das Spiel reichlich einfältig. Da hätte wohl jeder mehr erwartet. Vielleicht sollte man deshalb lieber auf das nächste Buch von WERNER warten.


C&C
Computers and Communications
NEC Drucker

Ausdauernd im Härtetest.



Der P9 XL von NEC scheut keine Herausforderung.

Favorisiert durch 16 Kilo **Präzision** und **Perfektion.** Und mit Doppelschacht-Einzelblatteinzug für die erste Seite und Folgeblätter beim Briefeschreiben erhöht sich das „Kampfgewicht" auf solide 20 Kilo. Dabei ist der P9 XL kein Sprücheklopfer, sondern ein **Meister seines Fachs.** Über die volle Distanz. Mit allen Feinheiten und Finessen, die einen wahren Champion ausmachen: Sein **24-Nadel-Präzisionsdruckkopf** garantiert auf jeden Fall den Sieg nach Punkten. Exakt und absolut gradlinig seine Druckqualität. Nahezu unschlagbar die **Buchdruckqualität** durch Carbon-Farbband. Begeisterung kommt auf, wenn es um Grafik geht. Brillante Leistungen und **hochaufgelöste Grafiken** aus der roten, blauen und gelben Ecke zeigen Wirkung bei harten Geraden und raffinierten Haken. Und Kondition hat der P9 XL wahrscheinlich mehr, als Sie sich vorstellen können. Härteste Einsätze steckt er weg, ohne auf die Bretter zu gehen. Beim NEC-Fachhändler sehen Sie diese **Höchstleistungen** schwarz auf weiß und in **Farbe.** Allerdings sollten Sie beim Kauf in Ihrem eigenen Interesse auf die **Original-NEC-Seriennummer** achten.



Klausenburger Straße 4, 8000 München 80, West Germany
Tel.: 089/93006-0, Telefax: 089/937776/8, Telex: 5218073 und 5218074 nec m

1st Proportional

Blocksatz und Proportionalschrift friedlich vereint



Bild 1: Das Optionsmenü zur Druckausgabe bei "1st Proportional"

Einen Schönheitsfehler von "1st Word" und "1st Wordplus" haben bisher viele Anwender zu Recht kritisiert: Es ist nicht möglich, Texte in Proportionalschrift und Blocksatz auf dem (Typenrad-) Drucker auszugeben. Abhilfe schafft jetzt ein neues Druckprogramm: "1st Proportional".

Wer mit "1st Word" oder "1st Wordplus" arbeitet, wird bei der Druckeranpassung schnell festgestellt haben, daß beide Programme die schönste aller Schriftarten, die Proportionalschrift, nicht im Blocksatz auf Papier bringen können. Das liegt daran, daß "1st Word" von einem äquidistanten Zeichensatz ausgeht, bei dem alle Buchstaben die gleiche Zeichenbreite besitzen. Bei einem proportionalen Zeichensatz variiert hingegen die Buchstabenbreite in Punktschritten, so daß z.B. ein "i" viel schmaler gedruckt wird als ein "m".

Schweizer Käse in Schwarz

Auch die integrierte Graphikausgabe ist bei "1st Wordplus" kein „Plus"punkt. Die englische Textverarbeitung war für die immer seltener eingesetzten 9 Nadeldrucker konzipiert. Wer inzwischen über einen Schönschriftdrucker mit 18 oder 24 Nadeln verfügt, kann mit dem "Snapshot" erstellte Graphiken nur in einer 8 Nadel-Emulation zu Papier bringen, — das sieht aus wie Schweizer Käse in Schwarz und meistens stimmen Bildschirmbild und Druckbild nicht überein.

Alle Möglichkeiten dieser Welt...

Beide Probleme löst "1st Proportional" auf höchst elegante Weise und bietet dazu noch einige Features, die vorbildhaft sind. "1st Proportional" übernimmt einen von "1st Word" oder "1st Wordplus" erstellten Text, berechnet die "Microspaces" zwischen zwei Wörtern und gibt ihn in Proportionalschrift und Blocksatz auf dem Drucker aus. Optional können alle Snapshotgraphiken in höchster Qualität ausgegeben werden, die Rückseiten lassen sich selbst bei Verwendung eines Einzelblatteinzuges bedrucken und die Kopf- und Fußzeilen kann man auf Wunsch unterstreichen und fett wiedergeben.

Doch zunächst zum Programm selbst: "1st Proportional" wird auf einer einseitig formatierten Diskette mit mehreren Druckertreibern (auch für Typenraddrucker!), Beispieldokumenten und ausführlicher Anleitung geliefert. Das kompilierte GFA-Basic Programm nimmt rund 80 KByte Speicherplatz ein und kann mit einem Doppelklick gestartet werden. Auf einen hinderlichen und lästigen Kopierschutz hat der Autor verzichtet; — dafür wurde eine Seriennummer implementiert, die mit dem Namen des Anwenders in der Copyrightmeldung erscheint.

Im *Dateimenü* kann im Unterpunkt *„einen Text drucken"* der auszugebende Text mit der bekannten Objektauswahlbox angeklickt werden. Auf dem Bildschirm erscheint dann ein Optionsmenü, das in Bild 1 wiedergegeben wurde. Zunächst läßt sich — wie bei "1st Wordplus" — der Druck ab/bis Seite, der linke Rand, die Seitennummervorgabe und die Seitenvertauschung von Kopf- und Fußzeile einstellen. Desweiteren erwartet "1st Proportional" einen Eintrag bei „Faktor Buchstabenbreite". Das ist ein Korrekturfaktor der angibt, auf wieviel Prozent der äquidistanten Schrift die schmalere Proportionalschrift gestaucht werden soll. Ist die Option „Beidseitig drucken" eingeschaltet, werden zunächst alle Vorderseiten des Textes ausgedruckt. Anschließend erscheint eine Meldung, die bedruckten Blätter so einzuspannen, daß im nächsten Durchgang die Rückseiten bedruckt werden können.

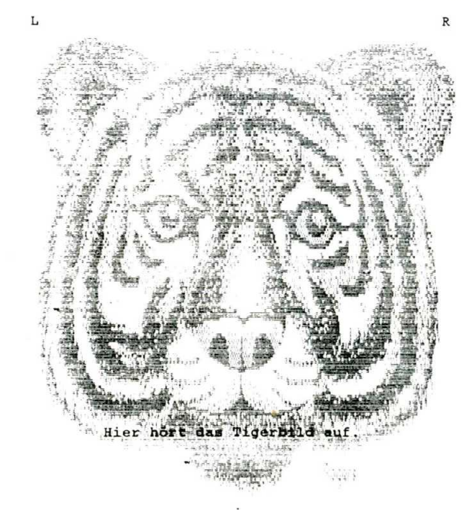
Schließlich kann man die Kopf- und Fußzeilen auch in Fettschrift und Unterstrichen ausdrucken lassen, um diese als Statuszeilen deutlicher hervorzuheben: eine sinnvolle Option, die leider bei "1st Word" fehlt. Wer mit "1st Wordplus" und dem "Snapshot" Accessory einige Bilder oder Graphiken in den Text eingefügt hat, kann wählen, ob diese mit ausgedruckt werden sollen oder nicht. Damit alle Einstellungen in der nächsten Arbeitssitzung stimmen, kann man unter *„Optionen sichern"* die gewünschte Konfiguration auf Diskette abspeichern, — sie wird beim nächsten Start automatisch mitgeladen.

Das Untermenü *„mehrere Texte drucken"* ist ein besonderes Bonbon für Buchautoren oder alle Vielschreiber, die kapitelweise ihr „opus magnus" verfassen. Es ermöglicht den verketteten Ausdruck von bis zu

1st PROPORTIONAL

Einen Schönheitsfehler von "1st Word" und "1st Wordplus" haben bisher viele Anwender zu Recht kritisiert: Es ist nicht möglich, Texte in Proportionalschrift und Blocksatz auf dem (Typenrad-) Drucker auszugeben. Abhilfe schafft jetzt ein neues Druckprogramm: "1st Proportional".

Wer mit "1st Word" oder "1st Wordplus" arbeitet, wird bei der Druckeranpassung schnell festgestellt haben, daß beide Programme die schönste aller Schriftarten, die Proportionalschrift, nicht im Blocksatz auf Papier bringen können. Das liegt daran, daß "1st Word" von einem äquidistanten Zeichensatz ausgeht, bei dem alle Buchstaben die gleiche Zeichenbreite besitzen. Bei einem proportionalen Zeichensatz variiert hingegen die Buchstabenbreite in Punktschritten, so daß z.B. ein "i" viel schmaler gedruckt wird als ein "m". Auch die integrierte Graphikausgabe ist bei "1st Wordplus" kein "Plus"punkt. Die englische Textverarbeitung war für die immer seltener eingesetzten 9 Nadeldrucker konzipiert.[1]



Die Markierungen geben an, wo unser Tiger auf dem Bildschirm erscheint. Da 1st Proportional bildschirmgerecht arbeitet, ist auch die nachträgliche Beschriftung von Graphiken kein Problem mehr.

Das Bild wurde übrigens von der zweiten hellblauen Originaldiskette des Programmes 1st Wordplus übernommen.

[1] Wer inzwischen über einen Schönschriftdrucker mit 18 oder 24 Nadeln verfügt, kann mit dem "Snapshot" erstellte Graphiken nur in einer 8 Nadel-Emulation zu Papier bringen, - das sieht aus wie Schweizer Käse in Schwarz und meistens stimmen Bildschirmbild und Druckbild nicht überein.

Bild 2: Ein Text ausgedruckt mit "1st Wordplus" ...

25 Texten. Jedem der eingetragenen Texte können wahlfrei und unabhängig voneinander im Optionsmenü beliebige Optionen zugeordnet werden. Die eingetragenen Texte können mit den eingestellten Optionen auf Diskette gespeichert bzw. wieder geladen werden. Weiterhin kann beim Ausdruck mehrerer Texte die Seitennummer kontinuierlich durchgezählt werden, ohne daß die Seitenzahl der Texte bekannt sein muß.

Im *Graphikmenü* findet man drei Auswahlmöglichkeiten. Die erste und zweite Graphikbreite ergeben ein formatgerechtes Bild auf dem Papier, das exakt dem Monitorbild entspricht. Ein Unterschied besteht nur in der Zeilenbreite des Bildes. Nahezu einmalig ist jedoch der dritte Modus *„Graphik unverzerrt"*. Während fast alle Malprogramme im Ausdruck einen Kreis zur Ellipse stauchen, ermöglicht "1st Proportional" in dieser Einstellung die Graphikausgabe mit unveränderten Seitenverhältnissen, — da bleibt die Tortengraphik wirklich rund.

Druckeranpassung leicht gemacht

Bevor jetzt der eigentliche Ausdruck beginnen kann, muß zunächst eine eigene Druckeranpassung für "1st Proportional" erstellt werden. Glücklicherweise hat der Programmautor die wesentlichen Prinzipien der "1st Wordplus" Anpassung übernommen, so daß diese lästige Prozedur einfach und schnell zu erledigen ist. Zunächst wird mit einem Editor ein ASCII File erstellt, das aus einer Steuerzeichentabelle (für das Einschalten aller Textattribute) und einer Buchstabentabelle (für die Zeichenfilterung) besteht. Zusätzlich muß zu jedem Buchstaben die Zeichenbreite (für den korrekten Randausgleich) angegeben werden. Für die Graphikansteuerung muß zum einen angegeben werden, ob man mit einem 9, 18 oder 24 Nadeldrucker arbeitet und zum anderen die Initialisierungssequenz zum Umschalten in den Graphikmodus des Druckers. Das erstellte ASCII File wandelt "1st Proportional" anschließend im Menüpunkt *Drucker* in eine kürzere Parameterdatei um, die bei jedem Start automatisch eingelesen wird.

Zur Ermittlung der Buchstabenbreite, die in vielen Druckerhandbüchern entweder gar nicht oder bisweilen mit falschen Werten angegeben ist, hilft das Menü *Extras*. Hier läßt sich zunächst ein Probeausdruck aller Zeichen vornehmen, deren Breite man dann schnell messen und im Untermenü „Rechnen" auf die druckerabhängige Einheit umrechnen kann. Somit ist auch an alle Anwender gedacht, die nur über ein unvollständiges Handbuch verfügen.

Ein Schriftbild wie's im Buche steht...

Nachdem nun alle Anpassungsprobleme erledigt sind, muß sich "1st Proportional" an seinen Druckergebnissen messen lassen. Wir haben zum Vergleich einen identischen Text einmal mit "1st Wordplus" und zum anderen mit "1st Proportional" auf dem NEC P6 ausdrucken lassen. Das Ergebnis sehen Sie in Bild 2 und 3. Sofort augenfällig ist natürlich die sehr schöne Proportionalschrift in Blocksatz, die mit manchem Laserdrucker durchaus mithalten kann. Wenn Sie auf beiden Reproduktionen die Kopf- und Fußzeilen vergleichen, werden Sie sehen, daß diese mit "1st Proportional" viel besser zur Geltung kommen. Der Unter- bzw. Überstrich entspricht exakt der Textbreite. Die wiedergegebene Graphik mußte mit "1st Wordplus" in einer 8 Nadel-Emulation ausgedruckt werden. Das sieht zum einen nicht sehr schön aus (die schwarzen Flächen sind alles andere als deckend) und zum anderen stimmen Bildschirmbild und Druckbild nicht überein (das sehen Sie an den Textmarkierungen). "1st Proportional" hingegen liefert gesättigte schwarze Flächen und das Prinzip „What you see is what you get" wird auch bei Graphikern eingehalten. Die beiden Ausdrucke sprechen wohl für sich: "1sr Proportional" liefert sowohl in der Text- als auch in der

Graphikausgabe ein Schriftbild von bisher unerreichter Qualität.

Ein Zeitvergleich für die beiden Ausdrucke zeigte, daß "1st Proportional" bei der (aufwendigeren) Graphik etwas langsamer war als "1st Wordplus". Umgekehrt war es bei der reinen Textausgabe, — hier erwies sich "1st Proportional" als schneller.

Zum guten Schluß sollen zwei Männer von "1st Proportional" nicht unerwähnt bleiben. So verfügt das Programm über die Möglichkeit, "1st Wordplus" automatisch nachzuladen. Diese Option funktionierte leider nur dann, wenn sich das Textprogramm nicht in einem Ordner befand — dies ist für Festplattenbesitzer u.U. ärgerlich. In der neuen Version 2.01, die jetzt ausgeliefert wird, ist dieses Manko allerdings behoben. Desweiteren bleibt anzumerken, daß "1st Proportional" leider nur mit dem Monochrommonitor lauffähig ist. Ob eine Farbversion demnächst lieferbar sein wird, ist noch ungewiß.

Fazit: Das Programm "1st Proportional" kann allen Anwendern empfohlen werden, die mit "1st Word" oder "1st Wordplus" Texte und Graphiken in höchster Qualität ausdrucken möchten. Für 89 DM stellt "1st Proportional" eine überaus sinnvolle Investition dar.

Bezugsnachweis:

"1st Proportional" ist für 89 DM bei Knißsoft, Victoriastraße 9, 5100 Aachen zu beziehen

(MS)

1st PROPORTIONAL

Einen Schönheitsfehler von "1st Word" und "1st Wordplus" haben bisher viele Anwender zu Recht kritisiert: Es ist nicht möglich, Texte in Proportionalschrift und Blocksatz auf dem (Typenrad-) Drucker auszugeben. Abhilfe schafft jetzt ein neues Druckprogramm: "1st Proportional".

Wer mit "1st Word" oder "1st Wordplus" arbeitet, wird bei der Druckeranpassung schnell festgestellt haben, daß beide Programme die schönste aller Schriftarten, die Proportionalschrift, nicht im Blocksatz auf Papier bringen können. Das liegt daran, daß "1st Word" von einem äquidistanten Zeichensatz ausgeht, bei dem alle Buchstaben die gleiche Zeichenbreite besitzen. Bei einem proportionalen Zeichensatz variiert hingegen die Buchstabenbreite in Punktschritten, so daß z.B. ein "i" viel schmaler gedruckt wird als ein "m". Auch die integrierte Graphikausgabe ist bei "1st Wordplus" kein "Plus"punkt. Die englische Textverarbeitung war für die immer seltener eingesetzten 9 Nadeldrucker konzipiert.[1]

L R



-Die Markierungen geben an, wo unser Tiger auf dem Bildschirm erscheint. Da 1st Proportional bildschirmgerecht arbeitet, ist auch die nachträgliche Beschriftung von Graphiken kein Problem mehr.

Das Bild wurde übrigens von der zweiten hellblauen Originaldiskette des Programmes 1st Wordplus übernommen.

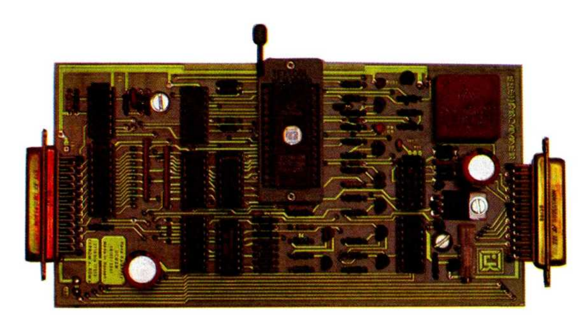
Hier hört das Tigerbild auf.

[1] Wer inzwischen über einen Schönschriftdrucker mit 18 oder 24 Nadeln verfügt, kann mit dem "Snapshot" erstellte Graphiken nur in einer 8 Nadel-Emulation zu Papier bringen, - das sieht aus wie Schweizer Käse in Schwarz und meistens stimmen Bildschirmbild und Druckbild nicht überein.

Bild 3: . . . und mit "1st Proportional". Vergleichen Sie selbst!

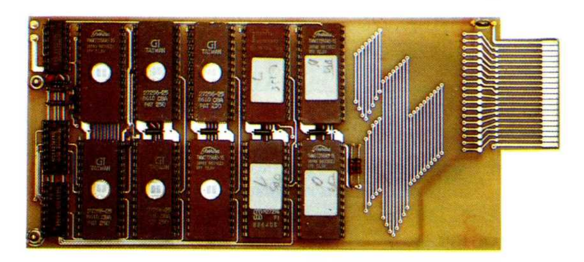
Aus der Hardwareküche der





Der Easyprommer

Ein universelles Programmiergerät für den ATARI ST, das nicht nur alle gängigen EPROMs der 27...-Serie brennt, sondern auch die modernen EEPROMs lesen und programmieren kann. Die sehr komfortable Software, natürlich voll GEM unterstützt, erlaubt alle nur denkbaren Manipulationen. Mit dem eingebuaten HEX/ASCII-Monitor lassen sich Änderungen der Daten blitzschnell realisieren. Der auf der Platine befindliche Druckertreiber erlaubt das problemlose Drucken der Daten. Zwei Zusatzprogramme ermöglichen das Brennen **jeder** nicht kopiergeschützten Software direkt von Diskette einschließlich Accessories und Autostart-Programmen. Der Easyprommer wird an den Druckerport angeschlossen und samt Diskette und Handbuch geliefert.



Die 576 KByte Easybank

Diese gebankte EPROM-Karte erlaubt es den ROM-Speicher des ATARI ST um über **570 Kilobyte** zu erweitern. Damit können Sie ganze Programmpakete samt Accessories sofort nach dem Einschalten parat haben. Selbst das Booten ist von dieser Karte problemlos möglich. Auf der Easybank finden bis zu acht EPROMs des Typs 27256 Platz, wobei die Steuersoftware automatisch in die ersten beiden EPROMs mitgebrannt wird. Die Platine wird mit vergoldeten Anschlüssen samt Diskette mit der benötigten Software und Anleitung geliefert.

Die 128 KByte Karte

Diese normale EPROM-Karte kann bis zu vier EPROMs des Typs 27256 aufnehmen. Sie eignet sich somit für kleinere Programme bzw. Accessories.

Verlängerungskabel zum Easyprommer

Mit diesem Kabel ist es möglich, den Easyprommer neben dem Rechner zu betreiben. Dieses Kabel ist notwendig, wenn der Easyprommer im Gehäuse geliefert wird.

EPROMs

Ohne EPROMs, nichts los! Wir liefern zu unseren Karten die passenden EPROMs..

27256 200ns (32 KByte EPROM)
27512 200ns (64 KByte EPROM)

Hiermit bestelle ich:

☐ Easyprommer (Fertiggerät) für DM 349,–

☐ Easyprommer (Bausatz) für DM 298,–

☐ 576 KByte EPROM-Karte (fertig bestückt o. EPROMs) für DM 148,–

☐ 128 KByte EPROM-Karte (fertig bestückt o. EPROMs) für DM 58,–

☐ Verlängerungskabel für Easyprommer für DM 39,90

____ Stück NMOS-EPROM 27256 à DM 15,–

____ Stück NMOS-EPROM 27512 à DM 26,–

Versandkosten: Inland DM 7,50 Ausland DM 10,– ☐ Vorauskasse
Auslandsbestellungen **nur** gegen Vorauskasse ☐ Nachnahme

Name: ____________ Vorname: ____________

Straße: ____________ Ort: ____________

Unterschrift: ____________

Bezugsquelle: ST Computer Redaktion
Schwalbacher Straße 64
D-6236 Eschborn
Tel. 0 61 96/ 48 21 58

PANIP setzt (alles) in Bewegung



Eine plötzlich aufkommende Briese



...läßt den bisherigen Surfer...



...noch kräftig vorbeiziehen

PANIP – Picture ANImations Programm – wie schon der Titel vermuten läßt, handelt es sich um ein Programm zum Erstellen von Bild-Animationen. PANIP ist, von der Bedienung her gesehen, sehr logisch aufgebaut, denn es folgt genau den Schritten, die bei der Erstellung eines Films notwendig wären: Hintergrund festlegen, Objekte definieren, filmen, Schnitt – fertig. Danach bleibt nur noch das Ansehen des fertigen Machwerks.

Verfolgen wir die Punkte im Einzelnen:

Als erstes wird der Hintergrund bzw. werden bis zu drei davon ausgesucht, die jedoch im NEOchrome-Format vorliegen müssen.

Der nächste Schritt ist die Erstellung der Objektliste, in der alle Objekte eingetragen werden, die später im Film bewegt werden sollen. Die Objekte können auf zwei Bildern verteilt sein, die dann eingeladen werden. Durch Einrahmen mit einer Box können nun bestimmte Teile der Bilder als Objekte definiert und in die Liste übernommen werden.

Als nächstes werden die Kameraeinstellungen vorgenommen. Die Nummer der Spur, die Anzahl der Bilder pro Sekunde und die Sekunden pro Szene werden eingestellt.

Wenn alle diese vorbereitenden Aktionen durchgeführt sind, wird es ernst – die erste Szene wird gedreht! Vor dem jeweiligen Hintergrund erscheint das gewählte Objekt. Es hängt praktisch am Mauszeiger und folgt somit jeder Bewegung der Maus. Wenn nun noch die linke Maustaste gedrückt gehalten wird, dann wird diese Bewegung des Objekts (bzw. der Maus) aufgenommen – einfacher geht es kaum.

Nachdem die erste Szene 'im Kasten' ist, kann sie beliebig oft angeschaut

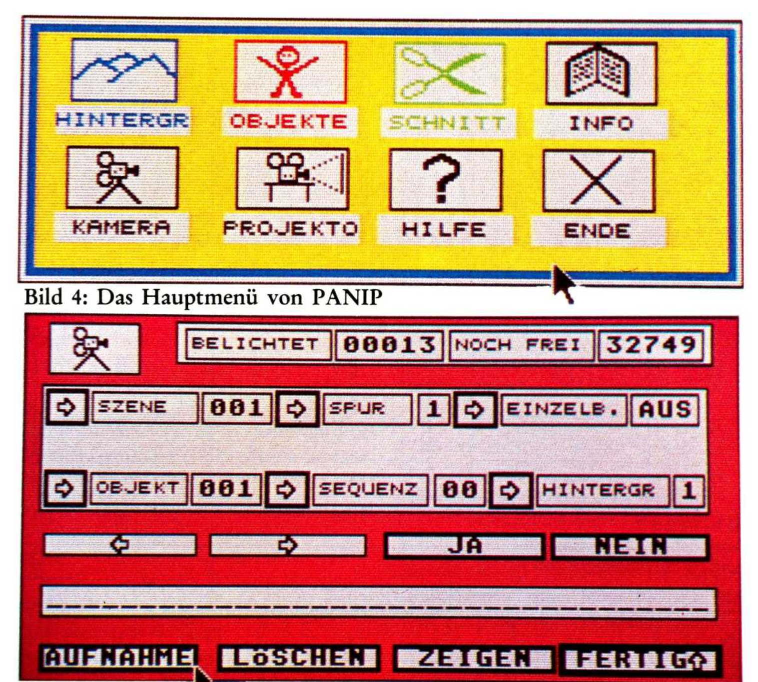


Bild 4: Das Hauptmenü von PANIP

Bild 5: Achtung: Aufnahme!

werden. Nun muß man entscheiden, ob man in diese Szenen noch ein Objekt einbringen will (die Spuren 2–7 sind ja noch frei), oder man geht zur nächsten über. Das Aufnehmen der zweiten Szene geht wieder genauso wie schon beschrieben. Einstellungen, Hintergrund und Objekt können nun natürlich variiert werden.

PANIP ist, wie aus dem Bericht hervorgeht, kein Malprogramm, sondern ausschließlich ein Animationsprogramm. Allerdings ist das dafür benötigte Malprogramm NEOchrome als Public Domain Software erhältlich. Ein echter Nachteil ist, daß nur dieses Format eingelesen werden kann. Zwar kann jedes Bild relativ leicht in ein anderes Format umgesetzt werden, aber das bedeutet einen gewissen Umstand, der nicht nötig gewesen wäre.

Mit PANIP läßt sich eine einfache Form der Animation verwirklichen, denn filmähnliche Sequenzen mit sich bewegenden Objekten sind nicht möglich. Mit PANIP können lediglich Bildausschnitte über den Bildhintergrund bewegt werden. Dies entspricht deshalb mehr einer Sprite- oder Shape-Bewegung als einer Animation wie sie z. B. mit dem AIGIS ANIMATOR oder dem FILMDIRECTOR möglich ist.

Ansonsten ist PANIP ein sehr leicht zu bedienendes Programm. Die Benutzerführung ist sehr gut aufgebaut und schnell zu erlernen. Alle Einstellungen und Auswahlen werden über Symbole gekennzeichnet, die so eindeutig sind, daß die Anleitung kaum notwendig ist.

Dafür muß man jedoch einige Einschränkungen in Kauf nehmen. Neben den bereits erwähnten Punkten ist auch die Beschränkung auf nur drei Hintergrundbilder bald recht lästig. Auffallend an der Qualität des fertigen Films ist, daß, wenn viele Objekte animiert werden, die einzelnen Bewegungen recht ruckartig werden. Dies kann sicherlich störend sein, allerdings reicht es für einfache Anwendungen meistens aus. Wer seine Erwartungen nicht zu hoch ansetzt und eher Wert auf eine einfache Bedienung legt, dem kann PANIP durchaus empfohlen werden.

MN

Vertrieb:
Adventure-Soft
Günter Möhle
Postfach 10 29
6452 Hainburg

Preis: DM 124,50

Einführung in FORTH Teil III

Nachdem im letzten Teil der Aufbau einer Doppelpunktdefinition im Vordergrund stand, geht es diesmal um Entscheidungen und Wiederholungen. Außerdem wird gezeigt, daß für das Programmieren in FORTH ein Editor unerläßlich ist.

Entscheidungen

Jede Programmiersprache bietet die Möglichkeit, die Ausführung von Befehlen von einer Bedingung abhängig zu machen. In FORTH werden solche Entscheidungen denkbar einfach realisiert. Einer Entscheidung in FORTH geht in der Regel ein Vergleich zwischen zwei Zahlen voraus. FORTH-83 stellt hierzu eine Reihe von Vergleichsoperatoren zur Verfügung. Ein solcher Vergleichsoperator ist z.B. '='. Um ihn anwenden zu können, müssen die beiden zu vergleichenden Zahlen zuvor auf dem Stack abgelegt werden:

12 34 ok

Aufgrund des Prinzips ("Last in — First Out"), nach dem der Stack in FORTH verwaltet wird, befindet sich die zuletzt eingegebene Zahl '34' an oberster Stelle im Stack und die davor eingegebene '12' eine Position darunter. Durch Eingabe von

= ok

wird der Vergleich durchgeführt. Für das Ergebnis, welches entweder wahr (beide Zahlen sind gleich) oder falsch (beide Zahlen sind ungleich) sein kann, wird ein entsprechendes Flag im Stack abgelegt. In FORTH-83 wird ein Wahrflag durch eine '-1' und ein Falschflag durch eine '0' dargestellt. Es gilt zu beachten, daß die beiden verglichenen Zahlen nach dem Vergleich vom Stack verschwunden sind. Sollten beiden Zahlen nach dem Vergleich für weitere Operationen benötigt werden, müssen sie vor dem Vergleich auf dem Stack kopiert werden. In unserem Beispiel befindet sich auf dem Stack nun ein Falschflag, da die beiden verglichenen Zahlen ja nicht gleich waren:

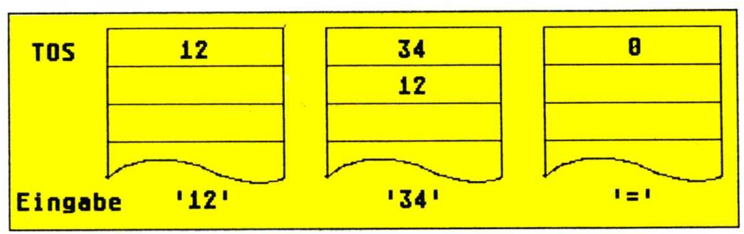


Abb. 1 Stackdiagramm 1

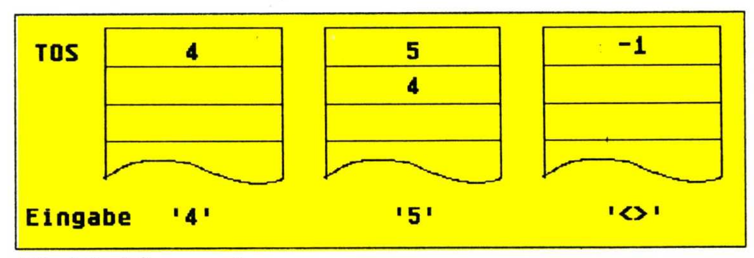


Abb. 2 Stackdiagramm 2

. 0 Ok

Abb. 1 zeigt den Zustand des Stacks nach den einzelnen Operationen. Neben '=' stellt FORTH-83 folgende Vergleichsoperatoren zur Verfügung:

< > 0= 0< 0>

Es fällt auf, daß ein '<>' Operator fehlt. Dieser läßt sich bei Bedarf jedoch leicht selbst definieren, wie die folgende Doppelpunktdefinition zeigt:

: <> = 0= ;

Die neue Wortdefinition wird durch einen Doppelpunkt eingeleitet. Dieser schaltet FORTH vom interpretierenden Modus in den kompilierenden Modus. (In volksFORTH-83 wird dies durch die Meldung „compiling" angezeigt) Alle nun folgenden Eingaben werden nicht mehr direkt ausgeführt, sondern in die neue Definition eingetragen. Danach folgt '<>', der Name der neuen Definition. Dem Wortnamen folgen die Worte, die beim Aufruf der neuen Definition ausgeführt werden sollen. Da wäre zunächst einmal '=', welches die beiden obersten Zahlen im Stack auf Gleichheit prüft und ein entsprechendes Flag auf dem Stack ablegt. Diese Flag wird vom nachfolgenden '0=' Operator gestestet. Bei diesem Vergleichsoperator wird die oberste Zahl im Stack mit Null verglichen. '0=' kehrt ein Flag in der obersten Speicherzelle des Stacks (dem "Top of Stack") um, da eine 0 im Stack in ein Wahrflag bzw. eine Zahl ungleich Null ist eine Null umgewandelt wird.

Bsp.

4 5 ok

<> ok

. -1 ok

Abb. 2 zeigt den Inhalt des Stacks während der Ausführung des Wortes.

Ein Vergleich macht allerdings noch keine Entscheidung. Schauen wir einmal, wie eine einfache Entscheidung in FORTH realisiert werden kann. Stellen Sie sich vor, Sie sollten ein Wort schreiben, welches

feststellt, ob die oberste Zahl im Stack kleiner als Null ist.

Der FORTH Ausdruck '0< IF ." ZAHL NEGATIV" THEN' prüft, ob die oberste Zahl im Stack kleiner als Null ist. Für den Fall, daß die Zahl kleiner als Null ist, wird die Anweisung zwischen IF und THEN ausgeführt. Ansonsten wird die Programmausführung bei dem ersten Wort nach THEN fortgesetzt. Dieser Ausdruck muß nun noch in eine Doppelpunktdefinition eingebaut werden, und das gesuchte Wort ist fertig:

```
: ?NEGATIV ( n —— ) 0<
   IF ." Zahl ist negativ" THEN ;
```

Das Wort '?NEGATIV' erwartet vor dem Aufruf eine Zahl auf dem Stack. Dies ist aus dem Wort selbst nicht unbedingt ersichtlich. Deswegen ist es sinnvoll, sich einer Notation zu bedienen, aus der hervorgeht, welche Parameter vor dem Aufruf eines Wortes auf dem Stack zu übergeben sind und welche Parameter sich nach der Ausführung eines Wortes auf dem Stack befinden. Die konsequente Anwendung einer solchen Notation, bei der der Stackinhalt vor und nach der Ausführung des Wortes in Klammern geschrieben wird, gehört einfach zu einem guten Programmierstil und verbessert die Lesbarkeit von Programmlistings erheblich. Bei der Eingabe der obigen Definition können Sie diese Stacknotation aber getrost weglassen, da sie auf die Funktion des Wortes keinerlei Einfluß hat.

Ungewohnt für FORTH Anfänger ist sicher der Aufbau einer 'IF ... THEN' Entscheidung und vor allem die Tatsache, daß die Bedingung vor dem Wort 'IF' aufgeführt wird. Dies liegt einfach daran, daß alle FORTH Worte ihre Parameter vor dem Aufruf auf dem Stack erwarten, auch 'IF' macht da keine Ausnahme. Der Aufbau einer Alternativentscheidung ist ebenfalls möglich, wie folgendes Beispiel zeigt:

```
: ?NEGATIV
   0<
   IF ." Zahl negativ" ELSE
      ." Zahl positiv" THEN ;
```

(die Meldung '?NEGATIV already exists' besagt lediglich, daß ein Wort mit diesem Namen bereits existiert)

Bsp.

-4 ?NEGATIV

Zahl negativ ok

55 ?NEGATIV

Zahl positiv ok

Alle Anweisungen zwischen 'ELSE' und 'THEN' werden ausgeführt, wenn die Bedingung vor 'IF' nicht erfüllt ist. Übrigens ist 'IF' nicht unbedingt auf einen vorherigen Vergleich angewiesen (außer Null natürlich) an oberster Stelle im Stack, damit der 'IF' — Teil ausgeführt wird — Probieren Sie es ruhig einmal aus.

Bliebe noch zu erwähnen, daß 'IF ... THEN' / 'IF ... ELSE ... THEN' Anweisungen beliebig verschachtelt werden können. Aus Gründen der übersichtlichkeit ist es allerdings ratsamer, ab einer bestimmten Verschachtelungstiefe auf eine Mehrfachentscheidung wie z.B. 'CASE ... OF' zurückzugreifen.

Programmieren in FORTH heißt den Sprachkern um neue Wortdefinitionen zu erweitern. Die Verarbeitung des Quelltextes (das ist der Oberbegriff für alles, was zur Ausführung gebracht werden soll) übernimmt der Textinterpreter. Grundsätzlich gibt es zwei Möglichkeiten, dem Textinterpreter den Quellkode zuzuführen. Einmal kann der Quelltext direkt über die Tastatur eingegeben werden. Alle Eingaben über die Tastatur (Zahlen, Befehle und auch Wortdefinitionen) werden unter dem Begriff Eingabestrom (engl. Input Stream) zusammengefaßt. Nach dem Betätigen der Return Taste wird der gesamte Eingabestrom, der im Eingabepuffer gesammelt wurde, vom Textinterpreter verarbeitet. Normalerweise wird jeder Befehl, der im Eingabestrom auftaucht, vom Textinterpreter sofort zur Ausführung gebracht (interpretierender Modus). Stößt der Textinterpreter allerdings auf einen ':', der ja bekanntlich eine Wortdefinition einleitet, werden alle weiteren Eingaben nicht mehr direkt ausgeführt, sondern in den Arbeitsspeicher (genauer gesagt in das Wörterbuch) kompiliert (kompilierender Modus). Stören Sie sich im Moment noch nicht an dem Begriff „kompilieren". Ich werde später dazu eine Definition nachliefern. Dieser kompilierende Modus kann z.B. durch ein ';' wieder aufgehoben werden. (Sie werden sich erinnern, die Doppelpunktdefinition wird durch ein ';' wieder beendet).

Die direkte Eingabe über die Tastatur weist aber zwei Nachteile auf. Zum einen wird der eingegebene Quelltext sofort umgesetzt und kann später nicht mehr gelistet, geschweige denn editiert werden. Zum anderen führt jeder Eingabefehler dazu, daß der Textinterpreter den kompilierenden Modus abbricht und in den interpretierenden Modus zurückkehrt. Alle bis dahin gemachten Eingaben der begonnenen Doppelpunktdefinition sind verloren. Es ist deswegen wesentlich effektiver, den Quellkode zunächst mit Hilfe eines Editors einzugeben und ihn gegebenenfalls auf Diskette zu speichern.

Der FORTH Editor

Bei einem Editor handelt es sich allgemein um ein Programm, welches die Eingabe, Verarbeitung und Abspeicherung von Programmtext (Quelltext) ermöglicht. Es existiert in FORTH kein "Standard Editor". Vielmehr sind die Bedienung und der Komfort des Editors von System zu System stark verschieden. Bevor der Editor des volksFORTH-83 3.7 vorgestellt wird, noch einige Erläuterungen zu dem Prinzip, nach dem FORTH den Massenspeicher verwaltet.

Der gesamte zur Verfügung stehende Massenspeicher (auf Diskette oder Festplatte) wird in log. Einheiten, sog. Blocks, eingeteilt. Ein Block umfaßt 1024 Bytes und ist sozusagen die kleinste Einheit, die vom Editor bearbeitet werden kann. Jeder Block wird über seine Blocknummer angesprochen. In einem Block kann man nun den Programmtext oder Daten, mit denen das Programm arbeiten soll, eingeben. Damit ein einzelner Block vom Benutzer bearbeitet werden kann, muß dieser Block zunächst in einen speziellen Bereich des Arbeitsspeichers, des Diskettenpuffer, geladen werden. Dies kann explizit durch den Benutzer veranlaßt werden (z.B. durch Eingabe von 'n Block', wobei n die Blocknummer darstellt), oder indirekt beim Aufruf des Editors. In volksFORTH-83 wird der Editor zum Bearbeiten eines Blocks durch Eingabe von

3 1

aufgerufen, wobei '3' die Nummer des zu bearbeitenden Blockes angibt. (Ignorieren Sie die Aufforderung „Geben Sie Ihre ID ein..." einfach durch Drücken der Return Taste). Nun erscheint ein leerer Rahmen, der aus 16 Zeilen mit jeweils 64 Zeichen besteht, auf dem Bildschirm (Sollte aus irgendeinem Grund kein leerer Rahmen erscheinen, so verlassen Sie entweder den Editor mit 'CTRL-F' und wählen Sie einen anderen Block oder überschreiben Sie einfach den Programmtext, der sich in diesem Block befindet). Zur Übung können Sie am besten einmal das Beispiel aus Abb. 3 eingeben. (Machen Sie sich zunächst keine Gedanken über die Bedeutung dieses Beispiels, wir werden später

noch darauf zurückkommen). Ist der vollständige Programmtext eingegeben, ist es sinnvoll, diesen auch abzuspeichern. Dies geschieht wieder durch 'CTRL-F'. Der gesamte Inhalt des Blocks wurde damit auf Diskette zurückgeschrieben.

Ein einzelner Block kann durch 'LOAD' interpretiert werden. So wird in unserem Fall durch

3 LOAD ok

(vorausgesetzt natürlich, Sie haben für die Eingabe des Programmbeispiels den Block mit der Nummer 3 ausgewählt) der Inhalt von Block 3 interpretiert. Mit WORDS können Sie sich davon überzeugen, daß das Wörterbuch tatsächlich um ein Wort mit dem Namen 'WARTE' ergänzt wurde. 'LOAD' macht eigentlich nicht viel mehr, als den Inhalt des angegebenen Blocks zuerst in den Diskettenpuffer zu laden (sofern er sich bereits dort befindet) und ihn dann dem Textinterpreter zu übergeben. Dieser behandelt die vom Diskettenpuffer kommenden Daten genauso, als würden sie direkt über die Tastatur eingegeben. Nach Eingabe von '3 LOAD' wurde der gesamte Inhalt von Block 3 interpretiert. Mit 'WORDS' können Sie sich davon überzeugen, daß das Wörterbuch tatsächlich um ein Wort mit dem Namen 'WARTE' ergänzt wurde.

Das Laden eines Blocks ist nicht auf einen einzelnen Block beschränkt. Als Alternative bietet es sich an, an das Ende eines einzelnen Blockes das Wort '—>' anzuhängen, welches den darauffolgenden Block ebenfalls lädt. Eine weitere Möglichkeit, mehrere Blöcke auf einmal zu laden, bietet 'THRU'. So werden beispielsweise durch ' 1 10 TH-RU' die Blöcke 1 bis 10 nacheinander von Diskette geladen und interpretiert.

Damit soll die Beschreibung des Editors an dieser Stelle beendet sein. Da der Edtior im Grunde ständig benötigt wird, werden Sie weitere Erläuterungen noch an mehreren Stellen im Laufe dieser Einführung finden.

Wiederholungen

Neben den Entscheidungen gehören die Wiederholungen zu den wichtigsten Elementen einer Programmiersprache. Eine einfache Wiederholungsanweisung haben Sie mit der 'DO ... LOOP' Schleife, bereits in der letzten Folge kennengelernt. Neben der 'DO ... LOOP' Schleife, bei der der Anfangs- und der Endwert vor dem Aufruf auf dem Stack abgelegt werden, müssen die Anzahl der Wiederholungen also im voraus feststeht, ist dies bei der Wiederholungsanweisung 'BEGIN ... UNTIL' nicht der Fall. Die Anzahl der Durchläufe hängt vielmehr von einer Bedingung ab. In Abb. 3 finden Sie ein Beispiel für eine 'BEGIN ... UNTIL' Anweisung. Sofern Sie den letzten Abschnitt durchgearbeitet und das Wort 'WARTE' mit Hilfe des Editors eingegeben und mit 'LOAD' kompiliert haben, können Sie 'WARTE' direkt aufrufen. Ansonsten holen Sie nun die Eingabe der Definition in Abb. 3 (entweder mit Hilfe des Editors oder direkt über die Tastatur) nach.

```
WARTE
BEGIN
  " DRÜCKE EINE TASTE"
 KEY?
UNTIL  ;
```

Abb. 3

Die Anweisungen innerhalb von 'BEGIN ... UNTIL' (die Ausgabe eines Textes bzw. das Prüfen einer Tasteneingabe durch 'KEY?') werden solange wiederholt, bis eine Taste gedrückt wird. Solange keine Taste gedrückt wird, liefert KEY? eine Null. Erst beim Betätigen einer Taste wird ein Wahrflag im Stack abgelegt. 'UNTIL' erwartet ein Flag im Stack, wie es z.B. von KEY? geliefert wird. Liegt ein Wahrflag an oberster Stelle im Stack, wird die Wiederholung abgebrochen, ansonsten läuft die Schleife ein weiteres mal durch. Die Anzahl der Wiederholungen hängt also in diesem Beispiel von dem Flag ab, welches von KEY? im Stack abgelegt wird.

Für viele Anwendungen ist die 'BEGIN ... UNTIL' Anweisung nicht ausreichend. So wäre es denkbar, daß eine Anweisung nur ausgeführt werden soll, während eine Bedingung erfüllt ist. Auch bei der 'BEGIN ... WHILE ... REPEAT' Anweisung hängt die Anzahl der Durchläufe vo neinem Flag ab, das sich vor der Ausführung von 'WHILE' an oberster Stelle im Stack befindet. Dieses Flag wird von 'WHILE' geprüft. Handelt es sich um ein Wahrflag, werden zunächst die Anweisungen zwischen 'WHILE' und 'REPEAT' ausgeführt und dann die Schleife ein weiteres mal durchlaufen. Falls sich vor der Ausführung von 'WHILE' ein Falschflag an oberster Stelle im Flag befindet, wird die Schleife abgebrochen. (War die Bedingung vor 'WHILE' bereits beim erstenmal nicht erfüllt, so werden die Anweisungen zwischen 'WHILE und REPEAT' kein einziges Mal ausgeführt.) Mit Worten ließen sich ‚BEGIN ... UNTIL' als eine „Tue solange bis eine Bedingung erfüllt ist" — und 'BEGIN ... WHILE ... REPEAT' als eine „Tue solange während eine Bedingung erfüllt ist" Schleife beschreiben.

Erwähnenswert ist schließlich noch die ‚BEGIN ... AGAIN' Schleife. Hierbei handelt es sich um eine sog. Endlosschleife. Abb. 4 zeigt ein einfaches Beispiel. Beim Aufruf von 'EWIG' werden solange Sternchen produziert, bis Sie entweder den Rechner neu starten oder bei Ihnen der Strom ausfällt. Solche Endlosschleifen haben durchaus ihre Berechtigung. So besteht beispielsweise der Textintepreter in FORTH, welcher ständig Eingaben von der Tastatur entgegennimmt und verarbeitet, im Kern aus einer solchen ‚BEGIN ... AGIN' Schleife.

```
EWIG
 BEGIN
  42 EMIT
 AGAIN ;
( EMIT gibt das Zeichen, dessen ASCII
Code sich an oberster Stelle im Stack
befindet aus )
```

Abb. 4

Konstanten und Variablen

Ein Thema, welches bislang noch gar nicht erwähnt wurde, sind Konstanten und Variablen. Allerdings auch nicht ganz grundlos, denn sowohl Konstanten als auch Variablen spielen in FORTH eher eine untergeordnete Rolle. Der Grund dafür liegt in der Tatsache begründet, daß die Parameter eines FORTH Programms in den meisten auf dem Stack verwaltet werden. Dennoch haben auch Konstanten und Variablen in FORTH ihre Berechtigung, da sie den Stack entlasten und vor allem die Lesbarkeit und Übersichtlichkeit von Programmen beträchtlich steigern — eine Notwendigkeit sind sie allerdings nicht.

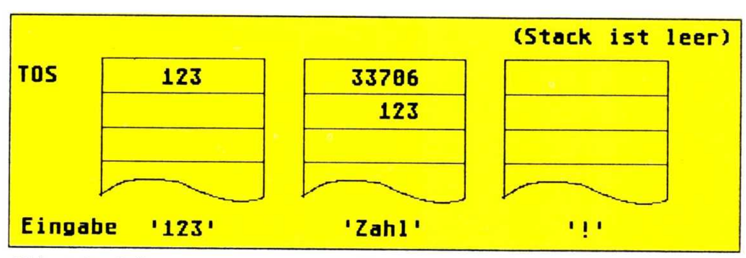


Abb. 5 Stackdiagramm

```
VARIABLE ZAEHLER
0 ZAEHLER !

: QUADRAT
  DUP *  ;

: ALLE_QUADRATE
   BEGIN
     ZAEHLER @ QUADRAT DUP .
     255 < WHILE
ZAEHLER @ 1+ ZAEHLER !
     REPEAT ;
```

Das Wort 'QUADRAT' ist ja noch aus der letzten Folge bekannt. Beachten Sie, daß das Ergebnis diesmal nicht ausgegeben wird, sondern auf dem Stack für weitere Berechnungen zur Verfügung steht. Die Sequenz 'ZAEHLER @ 1+ ZAEHLER !' liese sich besser durch die Sequenz '1 ZAEHLER +! ' ersetzen, wobei '+!' z.B. eine Variable, deren Adresse sich an oberster Stelle im Stack befindet um den Wert erhöht, der sich auf dem Stack darunter befindet.

Abb. 6 Auflösung der Übungsaufgabe

Ähnlich wie in PASCAL oder C müssen Konstanten bzw. Variablen vor ihrer Verwendung erst einmal definiert werden. Dies geschieht durch die Definitionsworte 'CONSTANT' bzw. 'VARIABLE' in der folgenden Form:

VARIABLE ZAHL ok

123 CONSTANT NOCH_NE_ZAHL

(Bei Variablen wird kein Initialisierungswert angegeben. Sie weisen daher zunächst irgendeinen Wert 0 auf). Bei Aufruf eines durch CONSTANT definierten Wortes wird der Wert der Konstanten an oberster Stelle im Stack abgelegt:

NOCH_NE_ZAHL <Return>

. 123 ok

Bei Variablen sieht es ein wenig anders aus. Beim Aufruf einer Variablen erscheint im Stack allerdings nicht der Wert der Variablen, sondern vielmehr die Adresse, unter der dieser Wert zu finden ist. Dementsprechend erhalten wir nach Eingabe von 'ZAHL':

ZAHL . -31830 ok

nicht den Wert der Variablen 'ZAHL', sondern die Adresse, unter der dieser Wert zu finden ist. Wie wir in der nächsten Folge sehen werden, wird eine Variable in einer ähnlichen Weise wie eine Doppelpunktdefinition in das Wörterbuch eingetragen. Die Adresse der Variablen 'ZAHL' hängt daher von der Anzahl und dem Umfang der davor gemachten Definitionen ab. Bevor Sie sich jetzt den Kopf darüber zerbrechen, wieso eine negative Zahl ausgeben, versuchen Sie es einfach noch einmal aber mit 'U.' anstelle von '.':

ZAHL U. 33706 ok

Der Grund für dieses unterschiedliche Verhalten liegt einfach darin, daß '.' nur vorzeichenbehaftete 16 Bit Zahlen ausgibt. Jede Zahl größer als ‚32768' wird daher mit einem Minuszeichen ausgegeben. Anders bei 'U.', welches jede Zahl als 16 Bit Zahl ohne Vorzeichen ausgibt.

Um auf den Inhalt einer Variablen zugreifen bzw. einen Wert in einer Variablen speichern zu können, benötigen wir ein Wort, welches den Inhalt einer beliebigen Speicherzelle in den Stack holt bzw. ein Wort, welches die umgekehrte Operation durchführt und einen Wert in einer beliebigen Speicherzelle ablegt. FORTH bietet dazu die Worte ' ' bzw. '!' an. (Auch wenn es sich hier nur um einzelne Zeichen handelt, werden auch ' ' bzw. '!' als Worte bezeichnet) Um mit diesen beiden Worten arbeiten zu können, werfen Sie zunächst einmal einen Blick auf das Stackverhalten der beiden Worte:

! (n addr ——)

(addr —— n)

Um '!' sinnvoll anwenden zu können, muß sich sowohl die Adresse der Variablen, als auch der abzuspeichernde Wert im Stack befinden und zwar in der angegebenen Reihenfolge. Durch eingabe von

123 ok

wird zunächst der Wert, der der Variablen zugewiesen werden soll, im Stack abgelegt und durch

ZAHL ok

schließlich die Adresse der Variablen. Die Eingabe von

! ok

erledigt den Rest und speichert '123' in der Variablen 'Zahl' ab. Es sei an dieser Stelle daraufhingewiesen, daß FORTH hier, wie auch in fast allen anderen Situationen keinerlei Überprüfung bezüglich der Korrektheit von Daten durchführt. Es liegt also an Ihnen, dafür zu sorgen, daß Daten auf dem Stack auch einen Sinn ergeben. Sollten Sie im obigen Beispiel etwa aus Versehen die Reihenfolge der beiden Zahlen vertauschen (was am Anfang sicher häufig der Fall sein wird), so wird '!'

die Adresse als die zu speichernde Zahl als die Adresse auffassen. '!' macht nichts anderes, als die zweitoberste Zahl im Stack unter der Adresse, die sich an oberster Stelle im Stack befindet, abzuspeichern. (siehe auch Abb. 5)

Nun wird sich sicher der eine oder andere Leser die Frage stellen, warum das alles so umständlich geht. Schließlich wäre es doch wesentlich naheliegender, eine Variable in der Form ' ZAHL = 123' zuzuweisen. Die Antwort lautet: selbstverständlich wäre dies auch in FORTH möglich. Nur würde dafür (unnötige) Rechenzeit geopfert werden, die zu Lasten der Performance geht. Gerade diese, zunächst vielleicht ein wenig merkwürdig anmutende Schreibweise, führt dazu, daß in FORTH Probleme in vielen Fällen eleganter und vor allem schneller gelöst werden können als in anderen Sprachen.

Zum Abschluß dieser Einführungsfolge noch eine kleine Übungsaufgabe. Schreiben Sie ein Programm, welches alle Quadratzahlen bis max. 255 ausgibt. Wenn Sie diese (und auch die letzte) Folge aufmerksam durchgearbeitet haben, dürfte Ihnen die Lösung dieses Problems nicht allzu schwer fallen. Die (oder besser gesagt eine mögliche) Lösung finden Sie in Abb. 6.

(PM)

LISP
mit pascalischer Abstammung

Eingebaute Funktionen in
LISPAS II

ABS, ADD1, AND, APPEND, APPLY, ASSOC, ATOM, *BYE, BREAKOFF, BREAKON*, CAR, CDR, CAAAR...C___R...CDDDR, *CLEARWINDOW, CLOCK, CLOSEWINDOW*, CONCAT, COND, CONS, *CONSP, CONSOLE, CURSORX, CURSORY, CUTOUT,* DEFUN, *DATEANDTIWE, DELETEWINDOW*, DIFFERENCE, DIVIDE, EQ, EQUAL, <>, *ERROR*, EVAL, FIRST, *FREE, FREQUENCY, FUNCTIONP*, GET, *GOTOXY, GREATERP, INTERACT*, INTERSECTION, *KEYSTROKE*, LAMBDA, LAST, LENGTH, *LESSP,* <=, LIST, LISTP, *LITERALP, LITERALS, LITLEN, LITREAD, LOAD, MAKEMENU*, MAPC, MAPCAR, MEMBER, *MENUCHECK, MENUENABLE, MINUS, MOD*, NCONC, NOT, *NTH,* NULL, NUMBERP, *NUMVAL, OKBOX, OPENWINDOW,* OR, *POSTMORTEM, PLUS,* PUT, *PRINTER*, QUOTE, *RANDOM*, READ, REMOVE, *REPEAT,* REVERSE, REST, RPLACA, RPLACD, *SASSOC, SAVE*, SECOND, SET, SETQ, *SETQQ, SEXREAD*, SORT, *SPACE, STACK, SUB1, SUBSET*, SUBST, *TABWIDTH, TEXTWINDOW, THIRD, TIMES,* UNION, *UNTIL, USING, VARP, WINDOWOPENP, WINDOWP, WINDOWS, WINDOWSIZE, WINDOWX, WINDOWY, WRITE, WRITELN,* ZEROP

Kursiv: Abweichungen vom COMMON LISP Standard (in Syntax oder Semantik)

LISP hat als eine der ältesten Computersprachen noch lange nicht ausgedient. Daran ändert auch das Erscheinen neuer Sprachen für den Bereich der KI, wie z. B. Prolog, nichts. Es ist also beileibe kein Anachronismus, wenn die Firma **Tommy Software** einen LISP-Interpreter für den ATARI ST anbietet. Über den Leistungsumfang und die Zielgruppe dieses Produktes informiert dieser Bericht.

Der Lieferumfang

LispasII wird mit einer schreibgeschützten Diskette und einem 57-seitigen Handbuch geliefert. Der Preis beträgt 198,– DM. Für weitere 148,– DM wird Mitte des Jahres (nach mündlicher Auskunft) ein Lispas Tutor erhältlich sein. Und der ist angesichts der knappen Dokumentation auch erforderlich, denn das Handbuch ist nur als alphabetische Zusammenfassung der Lispas Funktionen (siehe Anhang 1) mit kurzer Erklärung zu gebrauchen. Auch hier verspricht Tommy Software bis Mitte des Jahres Verbesserung, wenn das neue Handbuch veröffentlicht wird. Daß die Diskette schreibgeschützt ist, ist verständlich, aber für HD-Besitzer ärgerlich. Aber bei einem System, das wie Lispas weitgehend ohne Dokumentation auskommt (tut es wirklich!), ist der andere Weg (Dokumentation auf rotem Papier) eben nicht möglich.

Zielgruppen

Ganz eindeutig sind Schüler und Studenten, die LISP lernen wollen, die Hauptzielgruppe dieses Produktes. Und für jene bietet Lispas vor allen Dingen einen hervorragenden LISP-like-Editor. Die Bedienung des Editors, wie des ganzen Systems ist etwas gewöhnungsbedürftig, aber nach durchstandener Gewöhnungsphase angenehm.

Bedienung des Systems

Nach dem Starten meldet sich Lispas mit einem Fenster. In diesem Fenster spielt sich der ganze Dialog ab. Es können direkt Funktionen eingegeben werden, wobei das ganze Fenster als Arbeitsplatz zur Verfügung steht. Die RETURN-Taste dient dabei zum Zeilenvorschub, bewirkt aber keinerlei Übersetzungsfunktion. Mit den Cursortasten kann im Text beliebig herumgefahren werden. Ist die Funktion vollständig und korrekt eingegeben, signalisiert man Lispas den Wunsch zur Evaluierung mit der **F1**-Taste. Dieses ungewohnte Verhalten steht zwar in der Bedienungsanleitung, ist von mir aber zunächst überlesen worden (vielleicht doch zu klein gedruckt?!). Der Versuch, die Evaluierung über das Schließen des Fensters zu erzwingen gelingt zwar, leider führte dieser Weg aber zu einigen unerwarteten und unliebsamen Reaktionen (z. B. war nach dem Befehl (PRINTER) nicht mehr auf (CONSOLE) umzuschalten). Sollten Fehler in der Funktion vorhanden sein, deckt sie das System mit einer detaillierten Fehlermeldung auf. Dann wird es höchste Zeit, den Editor zu laden. Das geschieht mit dem Befehl (LOAD 'EDIT). EDIT ist ein LISP Programm und ein markantes Beispiel für die Qualität dessen, wozu ein Lispas Programm fähig ist. Natürlich steht dem Anwender der Source Code zur Verfügung, so daß er die Flut von Fensteranwendungen, die bei Benutzung des Editors auf ihn niederprasselt, als Vorlage für eigene Anwendungen nutzen kann. Der Start des Editors geschieht mit (EDIT **Funktionsname**). Wird der Funktionsname weggelassen, öffnet sich ein Menü-Fenster, in dem alle vorhandenen (und nicht editierbaren) Funktionen angezeigt werden. Man bringt den Cursor auf die gewünschte Funktion und drückt wieder **F1**. In einem neuen Fenster erscheint dann die gewünschte Funktion, damit Änderungen an ihr vorgenommen werden können. Allerdings arbeitet der Editor nicht ganz fehlerfrei. Wird bei der Änderung Text über den Rand des Fensters geschoben, dann ist er offensichtlich verloren! Mit **GotoLisp** oder **QuitEditor** aus dem Menübalken kann dann in der LISP Interpreter zurückgegangen werden. Mir erscheinen die beiden Varianten eher verwirrend; für meinen Geschmack hätte ein Menüpunkt ausgereicht. Der erste Menüpunkt ist wohl für schnelles Austesten von Programmen gedacht, denn nach einem Fehler kehrt man mit dem Befehl (EDIT) direkt in die eben verlassene Funktion zurück. Mit dem eben beschriebenen Verfahren ist es eine Freude, eigene LISP Programme zu entwickeln. Der Benutzer braucht sich auch über die Strukturierung seines Programmtextes keine Gedanken zu machen. Egal wie spaghettihaft der Text eingegeben wird, immer erscheint er im Editor und beim Abspeichern auf Diskette in schönster strukturier-

ter Schreibweise. Das Abspeichern der gewählten Funktionen gestaltet sich allerdings weniger komfortabel als das Editieren. Man muß alle zu speichernden Funktionen von Hand in einer Liste zusammengetragen und als Argument an die SAVE Funktion übergeben. Lediglich die Auswahl des Files geschieht wieder GEM-like über die Fileselector-Box. Eigenwillig ist auch das Format des gespeicherten Files: Alle Funktionen werden von einem Klammerpaar eingefaßt, und zwischen den Funktionen ist kein Zwischenraum vorhanden. Will man vorhandene LISP-Programme unter Lispas laden, muß man sie erst in dieses Format bringen. Außerdem ist Lispas sensitiv für Groß-Kleinschreibung. D. h. Heinz, hEinz und heinz sind drei verschiedene Atome. Solange man sich angewöhnt, gleich zu Beginn einer Arbeitsphase die CAPS-LOCK Taste zu betätigen, ist das nicht weiter schlimm. Ich persönlich ziehe allerdings die in XLISP vorhandene Umsetzung auf Großschrift vor, weil die implementierten Funktionen ja doch nur in Großschrift erkannt werden.

Der Leistungsumfang

Wie man Anhang 1 entnimmt, enthält Lispas die wichtigsten Common Lisp Funktionen. Zusätzlich sind viele Funktionen für eine einfache Fensterverwaltung vorhanden. Man kann sich über den Sinn und Unsinn von übermäßiger Fensterlei streiten, unstreitbar scheint mir aber die Notwendigkeit nach Nutzung der hervorragenden Grafikmöglichkeiten des ATARI ST. Und genau diese werden von Lispas ebenso wenig unterstützt wie von XLISP. Ebenso gravierend ist die Beschränkung auf Integer Arithmetik. Einem Computer wie dem ST steht einfach eine Sprache, die nur ganze Zahlen als arithmetische Datenobjekte zuläßt, nicht mehr gut zu Gesicht. Da bei vielen LISP Implementationen heute bereits objektorientierte Eigenschaften zu finden sind, stünden auch Lispas objektorientierte Datenstrukturen gut zu Gesicht, fehlen aber vollständig. Und wie ist es mit der Geschwindigkeit? Ich habe in einem Benchmarktest die Ergebnisse der Vergleichsmessungen zusammengefaßt (siehe Artikel **Benchmarktest**). Der Hersteller gibt an, daß Lispas mit dem CCD-Pascal compiliert wurde. Da XLISP mit dem Lattice C-Compiler compiliert wurde, ist eine vergleichbare Geschwindigkeit zu erwarten, und das trifft ja auch zu. Natürlich hat XLISP bei der Bildschirmausgabe Geschwindigkeitsvorteile, da XLISP im TOS-Modus arbeitet und nicht die langsamen GEM-Routinen zur Bildschirmausgabe benutzen muß. Daß Lispas von dem Hersteller als Lernsystem betrachtet wird, erkennt man unter anderem daran, daß es nicht möglich ist, sogenannte standalone Anwendungen zu erstellen, d. h. Programme, die ohne den Lispas Interpreter lauffähig sind: Eine notwendige Voraussetzung für den kommerziellen Einsatz. Unter diesem Aspekt sollte das System insgesamt betrachtet werden.

Fazit

Lispas läßt sich am treffendsten als Lisp-Editor mit nachgeschalteten Interpreter charakterisieren. Und das ist keineswegs abwertend gemeint. Denn wer mit XLISP gearbeitet hat, wird sicher mehr als einmal beim Klammernzählen ins Schleudern gekommen sein. Die Fähigkeiten des Interpreters selbst sind dagegen nicht überwältigend. Weder von der Geschwindigkeit, noch vom Leistungsumfang her bietet der Interpreter Besonderes, mit Ausnahme der Fenstersteuerung. Wer also LISP lernen will (und das sollte man!), keine kommerziellen Anwendungen im Sinn hat, und wem die Bedienung des XLISP Systems zu kompliziert ist, der hat mit Lispas sicher eine gute Möglichkeit, der Sprache näher zu kommen. Noch ein Wort zum Preis. Natürlich gibt es das PD XLISP. Hat Lispas da überhaupt noch eine Berechtigung? Ich glaube schon. Zählt man nämlich den Preis für einen guten Editor und eine Menü-Shell zusammen, legt man auch über DM 100,– hin und hat noch nicht einmal einen Editor, der die spezielle LISP-Syntax berücksichtigt.

SALIX-Prolog

Während sich MPROLOG sowohl vom Preis als auch vom Leistungsangebot eindeutig an den professionellen Anwender richtet, ist mit dem SALIX Prolog ein preisgünstiges Prolog erschienen, welches über einige interessante Eigenschaften verfügt und zudem mit DM 198,– preisgünstig ist.

Lieferumfang

SALIX Prolog wird auf einer einseitig formatierten Diskette geliefert, zusammen mit einem 293-seitigen ausgezeichneten Handbuch. Die Diskette enthält keinen Kopierschutz, so daß einer Installation auf einer HD nichts im Wege steht. Dafür ist aber die Dokumentation „kopiergeschützt“, da auf rotem Papier gedruckt. Ich halte das durchaus für eine wirkungsvolle Art des Kopierschutzes, weil bei einem komplizierten System wie Prolog und solch ausgezeichneter Dokumentation wohl kaum jemand auf die Orginaldokumentation verzichten möchte. Natürlich setzt dieses System des Kopierschutzes eine hohe Qualität der Dokumentation voraus. Und die ist bei diesem Produkt gegeben. Auf 56 Seiten werden zunächst das Prinzip des Prolog Interpreters und dann die speziellen Eigenschaften dieser Implementation besprochen. Natürlich ist diese Einführung kein Ersatz für ein gutes Lehrbuch. Aber die wesentlichen Informationen über Ein-Ausgabe, arithmetische Prädikate usw. geben dem Prolog-Kenner einen schnellen Überblick über die Implementation. Es folgen 167 Seiten für die alphabetische Liste aller Systemfunktionen, mindestens eine Seite pro Prädikat, unterteilt in die Unterpunkte:

a) Syntax,

b) Funktion,

c) Anwendung,

d) FAIL,

e) ERROR,

f) Bemerkungen,

g) Abweichungen von Clocksin & Mellish Standard,

h) siehe auch, i) Beispiele.

Aus dieser Aufzählung dürfte bereits hervorgehen, wie sorgfältig die Dokumentation zusammengestellt wurde und wie sehr die Autoren bemüht waren, dem Standard von Clocksin &

Mellish [1] zu folgen. Auf diesen Seiten werden auch einige Prädikate aufgelistet, die nur in einem später erscheinenden SALIX Prolog Professional zu finden sind. Leider sind die Informationen über dieses Produkt in der Dokumentation dünn gesät. Es lag dem Autor auch keine Version dieses interessant klingenden Produktes vor. Nach dem ersten Eindruck der vorliegenden Dokumentation kann das SALIX Prolog Professional auch vom TOS gestartet werden, und besitzt ein Interface für C- und Assemblerroutinen. Die restliche Dokumentation widmet sich dem Debugging, den GEM-Funktionen, sowie dem alphabetischen Stichwortverzeichnis.

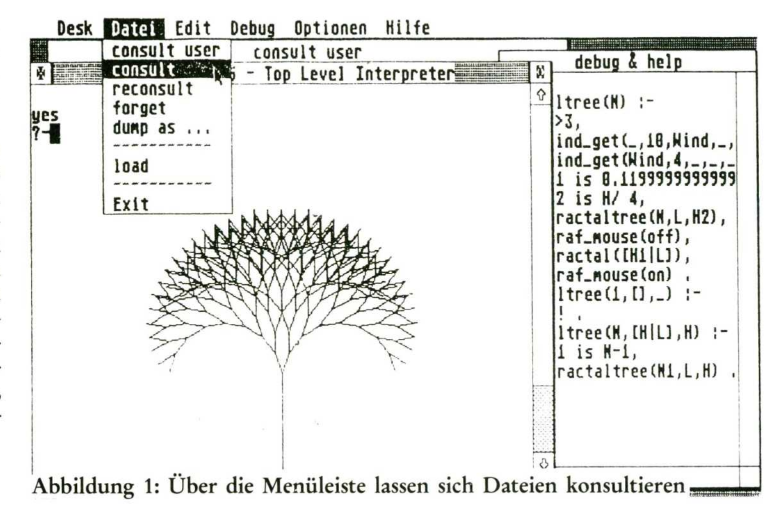


Abbildung 1: Über die Menüleiste lassen sich Dateien konsultieren

Eigenschaften

SALIX Prolog ist voll in GEM integriert. Es stellt sich fensterorientiert dar und stellt dem Benutzer alle GEM Funktionen zur Verfügung. Der Benutzer befindet sich nach dem Start zunächst im TOP-LEVEL Interpreterfenster. Aus der Menüleiste heraus ist die Consultierung von Prolog Files möglich, das Starten des Editors, einzelne Prädikate können gelistet und Optionen eingestellt werden (Abb. 1). Leider ist die Entwicklung des Systems wohl etwas unter Zeitdruck geraten, denn einige Details sind nicht ganz wasserdicht. Zunächst einmal fällt auf, daß dem System kein eigener Editor mitgegeben wurde. Es wird vorausgesetzt, daß der Benutzer das Textsystem 1st_Word besitzt. Dieses ist im File **START.PRO** vorgesehen. Will der Benutzer einen anderen Editor benutzen, dann muß er das in diesem File ändern. Beim Aufruf von 1st_Word gibt es dann den ersten Bug zu entdecken: Bei Anwahl eines Files auf einem anderen als dem Default-Laufwerk meldet sich 1st_Word erneut mit einer File-Selector-Box auf dem Default Laufwerk. Schlimmer wird es, wenn über die Menüleiste ein File auf Laufwerk B: zur Consultation angewählt wird, während Laufwerk A: als Default eingestellt ist: ein simpler Absturz (nichts geht mehr!) ist die Folge. Das Arbeiten mit 1st_Word als Editor gestaltet sich dagegen angenehm, weil nach dem Verlassen des Textprogramms sofort Prolog zur Verfügung steht. Man muß also im normalen Betrieb den Prolog Interpreter nicht verlassen und hat so etwas wie eine Menü-Shell zur Verfügung. Die erwähnten Fehler scheinen darauf zu beruhen, daß eine Pfadänderung in der File Selector Box nicht wahrgenommen wird. Dieser simple Fehler sollte schnell entfernt werden können. Ein weiterer Fehler ist der Absturz bei Verwendung einer RAM-Disk. Offensichtlich kollidiert die Shell-Verwaltung mit der RAM-Disk, denn der Start des Prolog Interpreters aus der RAM-Disk produziert eine Serie von Bomben. Der Benutzer ist daher auf die Verwendung der Diskettenlaufwerke bzw. der Hard Disk angewiesen, was zusammen mit den oben erwähnten Fehlern die Bedienung als 2-Drive System umständlich macht.

Leistungsdaten

Die Geschwindigkeit des Interpreters ist hervorragend (siehe Benchmarktest). Dies gilt sowohl für das Consultieren von Prolog Texten als auch für die Ausführungsgeschwindigkeit von Programmen. Dies ist umso erstaunlicher, als die Implementation die komplette Edinburgh-10 Syntax unterstützt und darüber hinaus noch folgende Ergänzungen zur Verfügung stellt:

Zu den Datenobjekten von SALIX Prolog gehören auch Gleitkommazahlen. Es werden eine Reihe von arithmetischen Prädikaten für Gleitkommazahlen zur Verfügung gestellt. Dazu gehören u. a. die wichtigsten trigonometrischen Funktionen.

SALIX Prolog ermöglicht den vollen Zugriff auf die GEM-Bibliothek. Es wurden sogar einige Standardprädikate so erweitert, daß Zugriff auf Elemente von GEM-Objektbäumen möglich wurde. Beispiele hierzu sind die Prädikate *read, readatom* und *readtoken*, die die Angabe eines Elementes aus einem Objektbaum ermöglichen, über den die Eingabe erfolgen soll.

Im angekündigten Professional Prolog wird darüber hinaus der Zugriff auf Prolog-fremde Datenstrukturen in Aussicht gestellt, wie sie in C- und Assemblerroutinen genutzt werden könnten.

Debugging

Gerade für Anfänger ist es sehr wichtig, gute Debugginghilfen zur Verfügung zu haben. Einerseits können damit Fehler gefunden werden, andererseits ist es besonders bei einer Sprache mit eingebautem Backtracking nützlich, den Programmablauf verfolgen zu können, um mit der Arbeitsweise des Interpreters vertraut zu werden. Das vorliegende Produkt bietet erfreulicherweise reichlich derartiger Hilfen. Mit Hilfe des Menüpunktes **Options**, aber auch über die Tastatur (es können übrigens alle Menüpunkte konventionell über die Tastatur eingegeben werden!) können folgende Debugging Hilfen gesetzt werden:

trace
Hiermit wird ein ausführliches Protokoll aller aufgerufenen Prädikate und der durchlaufenen Ports angefertigt. Ausschalten durch **notrace.**

spy
Spione kann man selektiv auf eine oder mehrere Prädikate ansetzen. Es

Aladin

Der Atari ST als Macintosh-Enhancer

Aladin macht **Userträume** wahr. Denn Aladin bringt Ihre professionelle **Macintosh-Software** auf den Atari ST. So z.B: Ready Set GO, WriteNow, TurboPascal, FullPaint, MacWrite, MacPaint, MacDraw, MacDraft, Helix, Comic-Works, MSWorks, und und und............

Aladin bringt Ihre Macdokumente auch in angemessener Form zu Papier. Denn Druckeranpassungen für **FX80** und **NEC P6** sind bereits im Lieferumfang enthalten. Prima !!

In Sachen Software hat Aladin sowieso **die Nase vorne**: integrierte, resetfeste Ramdisk; deutsche Tastaturbelegung; ein- und doppelseitige Laufwerke und Ihre Mac-Sofware wird um 20% schneller

Achtung Desktopper: Diese Anzeige haben wir mit Aladin und Ready Set GO! entworfen, denn Aladin bringt mit 640*400 Punkten eine ganze DinA4 Seite in der Breite auf den Bildschirm.

Aladin gibt Ihnen die gleiche Betriebssicherheit und Professionalität, wie Sie sie von Ihrem Macintosh gewöhnt sind.

Na, neugierig geworden? Wir von ProficomP oder Ihr Fachhändler informieren Sie gerne. Anruf oder Postkarte genügt.

Ach ja, und der Preis ist so niedrig, daß wir Ihn fast vergessen hätten. Denn Aladin kostet nur DM 299.- SFR 299.- ÖS 2499.-

ProficomP GmbH Rappenbergstraße 18a, 7507 Pfinztal 1, Tel 0721/469229

ASTROLOGISCHES KOSMOGRAMM – Nach Eingabe von Namen, Geb.-Ort (geogr. Lage) und Zeit werden errechnet: Siderische Zeit, Aszendent, Medium Coeli, Planetenstände im Zodiak, Häuser nach Dr. Koch/Schäck (Horoskop-Daten m. Ephemeriden) – Auch Ausdruck auf 2 DIN A4 mit allgemeinem Persönlichkeitsbild und Partnerschaft **75,–**

BIORHYTHMUS zur Trendbestimmung des seelisch-/geistig-/körperlichen Gleichgewichts, Zeitraum bestimmbar – Ausdruck per Bildschirm und/oder Drucker mit ausführlicher Beschreibung über beliebigen Zeitraum mit Tagesanalyse. Ideal für Partnervergleich **56,–**

KALORIEN-POLIZEI – Nach Eingabe von Größe, Gewicht, Geschlecht, Arbeitsleistung erfolgt Bedarfsrechnung und Vergleich m. d. tatsächlichen Ernährung (Fett, Eiweiß, Kohlehydrate). Idealgewicht, Vitalstoffe, auf Wunsch Ausdruck. Verbrauchsliste für Aktivitäten **56,–**

GELD – 25 Rechenroutinen mit Ausdruck für Anlage – Sparen – Vermögensbildung – Amortisation – Zinsen (Effektiv-/Nominal) – Diskontierung – Konvertierung – Kredit – Zahlungsplan usw. **96,–**

GESCHÄFT – Bestellung, Auftragsbestätigung, Rechnung, Lieferschein, Mahnung, 6 Briefrahmen mit Firmendaten zur ständigen Verfügung (Anschrift, Konten usw., Menge/Preis, Rabatt/Aufschlag, MwSt., Skonto, Verpackung, Versandweg usw.) **96,–**

ETIKETTENDRUCK – bedruckt 40 gängige Computer-Haftetiketten-Formate nach Wahl und Auflagebestimmung, kinderleichte Gestaltung, Ablage für wiederholten Gebrauch **89,–**

BACKGAMMON – überragende Grafik, gänzlich mausgesteuert, ausführliche Spielanleitung, lehrreiche Strategie des Computers, in 6 Farben bzw. Grauabstufungen bei S/W **58,–**

Prg. für alle ST-Modelle – Exzellent in Struktur, Grafik, Sound – alle Prg. in Deutsch – alle Prg. S/W und Farbe

GLOBALER STERNENHIMMEL – zeigt aktuellen Sternenhimmel für Zeit + Ort nach Eingabe Anklicken eines Objekts gibt Namen aus, Anklicken eines Namens zeigt das Objekt blinkend oder im Sternbild verbunden. Lupe für Großdarstellung mit Helligkeiten. 'Wandern' simuliert Bewegung oder Drehung der Erde. **89,–**

FONT EDITOR unter DEGAS – 12 bekannte Schriftarten m. deutschem Zeichensatz **64,–**

CASINO-Roulett – Mit Schnellsimulation, Chancentest, Sequenzenverfolgung, Kassenführung. Häufigkeitsanalyse. Setzen d. Anklicken d. Chancen auf Tischgrafik **68,–**

usw. usw. – Fordern Sie mit Freiumschlag unsere Liste an!
Im Computer-Center oder bei uns zu obigen, unverbindlich empfohlenen Preisen + DM 3,– bei Vorkasse oder DM 4,70 bei Nachnahme



ADRESSEN **66,–**
BIBLIOTHEK **86,–**
LAGERARTIKEL **86,–**
INVENTUR **96,–**

I. Dinkler · Idee-Soft
Am Schneiderhaus 17 · D-5760 Arnsberg 1 · Tel. 0 29 32/3 29 47

▶LOGISTIX◀

Die beste Software-Idee, seit es Tabellenkalkulation gibt!!
Integriertes Progr. mit: Kalkulation – Grafik – Datenbank – Zeitplanung
Preis: 398, – DM. Fordern Sie ausführlichen Farbprospekt an.

Alpha Computer, 1000 Berlin 31 ★ Dataplay, 1000 Berlin 31 ★ Ihre Computerei, 1000 Berlin 42 ★ Computare Elektronic, 1000 Berlin 61 ★ PS Data, 2800 Bremen ★ Neumann Personalcomp, 2850 Bremerhaven ★ S & F Datentechnik, 2950 Leer ★ Coppytip, 3550 Marburg ★ Basis Computer, 4400 Münster ★ City Elektronik, 4600 Dortmund ★ CSF Computer, 4800 Bielefeld 1 ★ Rocke Computer, 5090 Leverkusen 3 ★ A T C , Computersysteme, 5540 Prüm ★ Buchh. Finke, 5600 Wuppertal 13 ★ Schmitt Computersysteme, 6000 Frankfurt/11 ★ Dutkowski Computer, 6036 Karben 4 ★ Heim Computertechnik, 6100 Darmstadt-Eberstadt ★ Landolt Computer, 6457 Maintal 1 ★ Elphotec, 6500 Mainz ★ Schmitt Computer, 6500 Mainz 1 ★ Computer Center (Köbler), 6800 Mannheim ★ Heidelberger Comp. Center, 6900 Heidelberg ★ Schmidt Computer, 7000 Stuttgart 1 ★ Kirchner Datentechnik, 7080 Filderstadt 4 ★ Fritz Seel, 7100 Heilbronn ★ Wallisser, 7100 Heilbronn ★ DM Computer, 7530 Pforzheim ★ BUS, 7730 Vs-Schwenningen ★ CDS, 7800 Freiburg ★ H I B , 8500 Nürnberg 21

Wenn Sie LOGiSTiX bei Ihrem Händler nicht finden, bestellen Sie es bitte direkt bei uns.

Programm & Design

Friedensstr. 14 · 5433 Siershahn · Tel. 0 26 23-12 20

werden dann nur diese Prädikate im Debug-Fenster ausgegeben. Ausschalten durch **nospy**. Das Prädikat **debugging** liefert eine Liste aller aktiven Spione und mit **nodebug** werden alle Spione gleichzeitig ausgeschaltet.

leash
Leashing entspricht dem Setzen von Breakpoints. Mit Hilfe von **leash** kann man zusammen mit **trace** oder **spy** veranlassen, an bestimmten Ports zu stoppen. Mit **unleash** kann der **leash** Befehl wieder rückgängig gemacht werden.

Weiterhin erhält der Benutzer die Möglichkeit, eigene Debug-Hilfen zu programmieren, indem er das Prädikat **debug** definiert, welches bei einem Fehler aufgerufen wird. Für Anfänger besonders hilfreich dürfte eine kurze Anleitung zum debuggen von Programmen sein, mit Hinweisen auf die von Anfängern am häufigsten gemachten Fehler.

Was fehlt?

Wie oben erwähnt, orientiert sich der Autor von SALIX-Prolog am Clocksin & Mellish Standard. Leider fehlt dennoch ein wichtiger Aspekt von PROLOG: die Grammatik. Mit Hilfe einer Grammatik lassen sich automatisch Regeln übersetzen, was sowohl für die Analyse formaler Sprachen als auch für den Compilerbau interessant ist. Näheres zur Verwendung von Grammatiken siehe [2]. Allerdings liefert Salix Prolog das Prädikat **tco/2**. Mit Hilfe dieses Prädikates ist es möglich, beliebige Grammatiken selbst zu übersetzen.

Kompatibilität zu TOY-Prolog

Viele Interessenten für Salix Prolog kommen möglicherweise aus dem Lager der TOY-Prolog Benutzer (PD). Für diese ist extra ein Startfile START.TOY in dem Ordner Namens PARTBOX enthalten. Kopiert man diesen auf die Diskette als START.PRO (Nennen Sie den Originalfile START.PRO vorher um!), dann *versteht* Salix Prolog Toy Prolog Programme. Leider sind in diesem Startup File doch noch Fehler enthalten. Beispielsweise übersetzt Toy Prolog grammatische Regeln (s. o.). Mit dem neuen Start File ist Salix Prolog nun tatsächlich in der Lage, grammatische Regeln zu lesen. Leider aber wird die Übersetzung nicht fehlerfrei vorgenommen. Nach erster vorsichtiger Schätzung dürfte der Fehler bei der Übersetzung nichtterminaler Symbole entstehen, wenn die rechte Seite der grammatischen Regel mehr als zwei nichtterminale Symbole enthält, so daß mehr als die Variablen S, S0, S1 erforderlich sind.

Zukünftige Entwicklung

Das SALIX Prolog wird zur Zeit in der Version 1.4 ausgeliefert. Besitzer älterer Versionen können diese zum Selbstkostenpreis beim Heim Verlag umtauschen. Eine verbesserte Version wird außerdem für das nächste Jahr in Aussicht gestellt. Diese soll einen Editor für den Arbeitsbereich, ein BLOCK – BLOCKexit Konstrukt enthalten sowie die Verarbeitung unendlicher Datenstrukturen ermöglichen. Auch hier ist wieder mit einer liberalen Updatepolitik zu rechnen.

Fazit

Bis auf kleine Mängel im Bereich des File Handling ist Salix Prolog ein ausgesprochen leistungsfähiges Produkt. Die ausgezeichnete Dokumentation und die hervorragenden Debug-Hilfen machen es einem Anfänger leicht, mit Prolog warm zu werden. Amateure (im besten Sinne!) sind wohl die Hauptzielgruppe dieses Interpreters, dessen Preis von DM 198,– bei der gebotenen Leistung mehr als gerechtfertigt erscheint. Die Professional Version wird (nach letzter mündlicher Information) von der Firma Brainware in Berlin für DM 498,– vertrieben.

Literatur

[1] Clocksin, W. F. & C. S. Mellish. Programming in Prolog. Springer Verlag Berlin Heidelberg New York, 1981.

BENCHMARK

Benchmark (engl.: Meßlatte) ist die Bezeichnung eines Tests, der die Leistungsfähigkeit von Software klären soll. Ein absolutes Maß der Leistungsfähigkeit kann damit natürlich nicht gemeint sein, weil jedes Softwareprodukt seine speziellen Schwerpunkte besitzt, die es von den Konkurrenten unterscheidet. Man ist im allgemeinen schon froh, wenn ein Benchmark Test eine relative Aussage über eine Palette von Produkten erlaubt. Dieser Test untersucht die Leistungsfähigkeit der vier preisgünstigen KI-Sprachen für den ATARI ST: LISPAS und XLISP für die traditionelle KI-Sprache, LISP sowie TOY und SALIX Prolog für die modernere Sprache PROLOG.

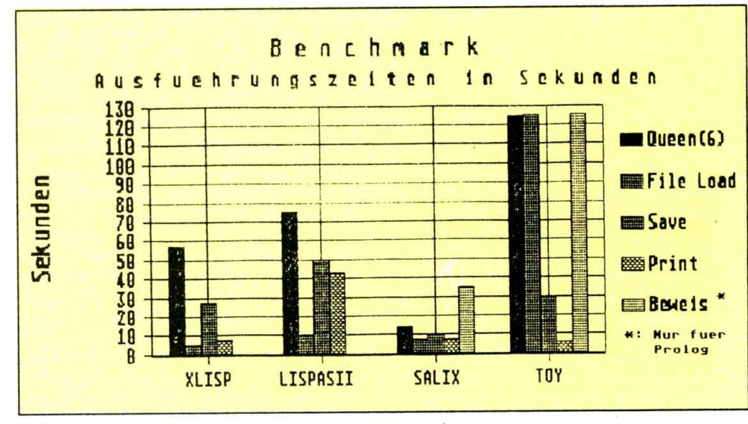


Abbildung 1: Benchmarktests der KI-Sprachen

Das Meßprinzip

Das Prinzip des Benchmark Tests ist einfach: man läßt ein bestimmtes Programm laufen und stoppt die erforderliche Ausführungszeit. Je kürzer die Ausführungszeit, desto leistungsfähiger das Programm. Oder? Ganz so einfach ist es natürlich nicht! Schließlich spielt auch noch eine Rolle, welchen

GEHEIM-Tips

für den Atari ST

copySTar v2.0

- Kopiert **alle** Atari ST Programme, **normale** und **kopiergeschützte**
- **Superschnelle Kopien** von 'normalen' Disketten in nur 32 Sekunden!!
- **Superschnelle Formatierung,** SS 16 sec
- Konvertiert in **Spezialformat** für **doppelte Geschwindigkeit** ohne zusätzliche Hardware
- **Update Service!** Jeder Kunde wird automatisch benachrichtigt
- Kinderleichte Bedienung durch **GEM**
- Ausführliches **Handbuch in Deutsch**
- Optionale **Erweiterung** der Diskettenkapazität **um 50 KB bzw. 100 KB**
- Kein Kopierschutz

Version 2.0 jetzt noch besser!

- **Testmöglichkeit der Drehzahl** des Laufwerkes
- **Überprüfung von Disketten** auf defekte Bereiche **(Qualitätstest)**
- Noch **bessere und erweiterte Kopierschutz-Erkennung**

nur DM 169,–

T.L.D.U. The Last Disc Utility

- Endlich eine **komplette Disk-Utility** mit allen Funktionen
- Beachten Sie z. B. den **Test in CP** 24/86
- Natürlich auch mit **Ram- und Harddisk**
- **Vollprogrammierbar** ('C'-ähnliche Macro-Sprache mit Editor)
- Echte (Sub-)Directory- und Fat-Funktionen
- 10 Buffer, **Disassembler,** Folge-Modus, **Suchen,** Vergleichen und mehr...
- **deutsche Handbuch**
- kein Kopierschutz

nur DM 149,–

Detective v1.0

- Erstelle **Assemblersource-Dateien** von jedem Maschinenprogramm
- Einfache GEM-Bedienung
- Fantastisch schnell
- Verwaltet Text-, Data-, BSS-Bereiche, **Symbole,** Marker und mehr
- Ein Muß für jeden ernsthaften Programmierer

nur DM 149,–

Wo? Natürlich bei Ihrem Atari-Händler! oder direkt bei STARSOFT Hannover

Versand in die ganze Welt!



Im Gosewinkel 20 · D-3000 Hannover 61

☎ 05 11 / 56 25 26

megaboard

Ein neues Großhirn für Ihren ST.

■ Megaboard ist die neue Speichererweiterung für alle ST-Modelle auf 2 oder 4 MByte.

■ Megaboard ist mit dem brandneuen 511000 bestückt – ein Beispiel für unsere Entwicklung und Leistungsfähigkeit auf dem ausschließlich neuesten Stand der Technik.

■ Megaboard hat seinen Platz im Originalgehäuse.

und ■ Megaboard wird von uns, d.h. durch geschulte Fachkräfte, eingebaut. – Das garantieren wir!

Rufen Sie uns an!

Tel: 06128/84734

oder schreiben Sie uns, und nutzen Sie unser aktuelles Angebot:

ECKL electronic
Erlenmeyerstraße 3
6204 Taunusstein

Fragen Sie auch nach unserer Speichererweiterung auf 1 MByte, und erkundigen Sie sich über unser Angebot von weiterem Zubehör.

ATARI ST® ist ein eingetragenes Warenzeichen der Atari-Corp. Sunnyvale, CA.

ECKL electronic

Einstieg in die Welt der Profis

Echtzeituhr

– Uhrzeit, Datum, Wochentag, Schaltjahr

Voll integriert – kein Ausgang wird belegt. Steckbar – kein Löten, keine Mechanikbearbeitung. Installationsdiskette liegt bei.

Echtzeituhr mit Lithiumbatterie **DM 126,–**

Epromkarten

256/128 KByte 4 Steckplätze **DM 79,–**

128/64 KByte 2 Steckplätze **DM 59,–**

Für 260 ST, 520 ST, 520 ST/M

Allen Geräten und Bausätzen liegt eine illustrierte, ausführliche Anleitung bei, die spezielle Kenntnisse überflüssig macht. Versand per NN.
Ausland Vorkasse + DM 8,50. Info kostenlos.
Händlerliste.

S. u. E. GmbH

Andreas Gerzen
Ing.-Büro für Datentechnik
Niederstraße 17
D-4019 Monheim
Telefon 02173/52200

1 M Byte Speichererweiterung

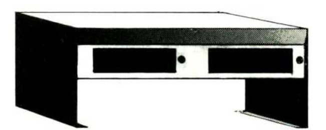
Einfacher, problemloser Einbau ohne Lötungen oder Blechbearbeitung. Einfacher, sicherer und vollständiger Funktionstest durch beiliegende G-S Routine auf Diskette. Äußerste Schonung des MMU-Sockels durch vergoldete Mikrosteckkontakte. Präzisionssockel. Extrem kurze Leitungswege. Selbstverständlich flimmerfrei.

Megaram, komplett, einbaufertig mit Prüfdiskette **DM 229,–**

Schluß mit dem Kabelsalat!

PC-Gehäuse

Bitte Info anfordern



Aufschiebbares Tischgehäuse zur Aufnahme von 2 Laufwerken, Netzteil, Erweiterungen, Steckdosen usw. Rechner unterschiebbar. Seitliche Tragegriffe.

DM 182,–

Sonderposten Gehäuse für 1 Laufwerk **DM 88,–**

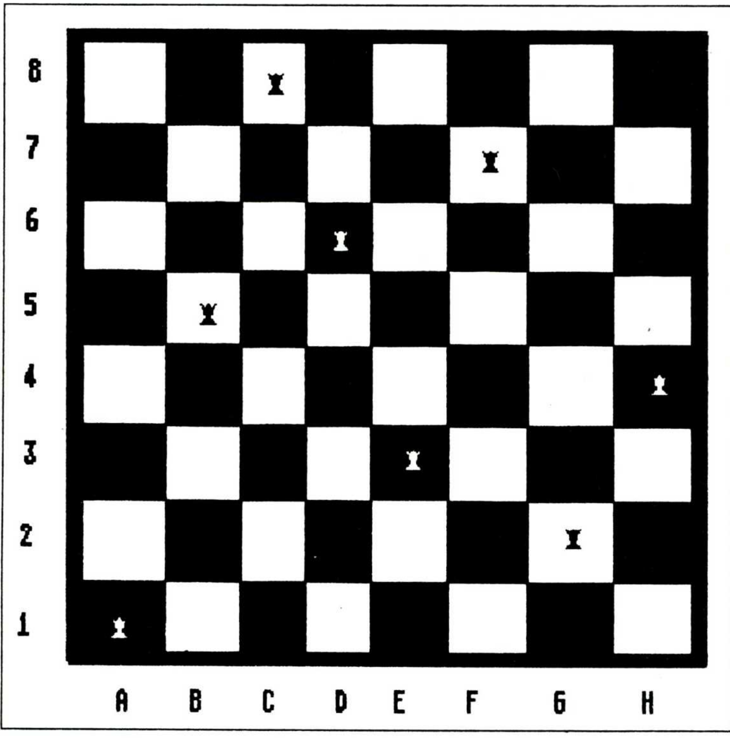


Abbildung 2: Eine Lösung des Damen Problems

```
((DEFUN BEDROHT (I J A B)
  (OR (= I A) (= J B) (= (- I J) (- A B))
   (= (+ I J) (+ A B))))
 (DEFUN KONFLIKT (N M BRETT)
  (COND ((NULL BRETT) NIL)
   ((OR (BEDROHT N M (CAAR BRETT) (CADAR BRETT))
     (KONFLIKT N M (CDR BRETT))))))
 (DEFUN QUEEN (GROESSE) (QUEEN-AUX NIL Ø GROESSE))
 (DEFUN QUEEN-AUX (BRETT N GROESSE)
  (COND ((= N GROESSE))
   (T (QUEEN-SUB BRETT N Ø GROESSE))))
 (DEFUN QUEEN-SUB (BRETT N M GROESSE)
  (COND ((= M GROESSE))
   (T
    (COND ((KONFLIKT N M BRETT))
     (T
      (QUEEN-AUX (CONS (LIST N M) BRETT) (+ N 1) GROESSE)))
    (QUEEN-SUB BRETT N (+ M 1) GROESSE))))
N (+ M 1) GROESSE))))))
)))))
```

Listing 1: Der Benchmarktest in LISP

Komfort das Programm liefert, wie fehlerfrei es ist und wie sicher eine Fehlbedienung abgefangen wird. Von der Dokumentation ganz zu schweigen. Trotzdem hat so ein simpler Benchmarktest seine Existenzberechtigung als zusätzliches Auswahlkriterium. Im Verlauf dieses Artikels sollen daher die Programme bzw. Methoden erläutert werden, die zu dem Ergebnis in Abb. 1 führen. Um die Ausführungszeit der verschiedenen Interpreter zu vergleichen, wurde das Damen-Problem in LISP und PROLOG programmiert und dessen Ausführungszeiten gemessen. Beide Versionen benutzen Listen zur Lösung des Problems. Hiermit ist ein Leistungsvergleich zwischen den beiden Sprachen überhaupt erst sinnvoll. Andererseits sollten auch spezielle PROLOG-Eigenschaften verglichen werden. Deshalb habe ich den Beweis **„Haßte Markus Cäsar?“** (siehe [1]), der ausgiebig von den Datenbankeigenschaften von PROLOG Gebrauch macht, zum PROLOG-internen Benchmark herangezogen. Ein vergleichbares Programm für LISP gibt es nicht (Jedenfalls nicht ohne das Listenkonzept). Für die praktische Arbeit mit den Sprachen ist die sogenannte *Turnaround*-Zeit interessant, das ist die Zeit, die man während der Programmentwicklung braucht, um ein Programm in den Interpreter zu laden, in den Editor zu geben und wieder zu laden. Ich habe mich darauf beschränkt, die Ladezeit und Speicherzeit eines ca. 3 kByte langen Programmes zu messen. Im letzten Test habe ich die Zeit gemessen, die ein Programm braucht, um Zahlen auf den Bildschirm auszugeben.

Queen

Das Damenproblem besteht darin, auf einem n★n-Schachbrett n Damen so zu plazieren, daß sie sich gegenseitig nicht bedrohen (Abb. 2 zeigt eine mit SALIX Prolog berechnete Lösung). Im Benchmark wurde auf eine grafische Darstellung verzichtet, weil in XLISP und LISPAS keine Grafik möglich ist. Listing 1 zeigt die LISPAS Version. Listing 2 die entsprechende PROLOG Version. Die Zeiten entsprechend der Lösung eines 6★6-Brettes. Näheres zum Problem und zur Lösung findet man bei [2] und [3].

Beweis

Das Programm zur automatischen Beweisführung findet man in dem Artikel [1]. Es wurde gemessen, wie lange der TOY Prolog (PD) bzw. SALIX Prolog Interpreter für den Beweis benötigen. Da dieses Programm die typische Prolog Struktur besitzt, wurde auf eine Implementation in LISP verzichtet.

File Save

Es wurde ein ca. 3 kByte langer Programmtext geladen. Im Fall von LISP handelt es sich um den PROLOG Interpreter, der auf der XLISP PD Diskette mitgeliefert wurde. Für Prolog diente wieder das Programm zur Beweisführung als Vorlage.

Load

Es werden die gleichen Programme

wie bei **File Save** benutzt. Die langen Lade- und Speicherzeiten erklären sich daraus, daß das Programm in ASCII-Text umgewandelt werden muß und umgekehrt.

Ergebnis

Wie die Abb. 1 eindeutig zeigt, ist SALIX Prolog der unzweifelhafte Gewinner dieses kleinen Benchmarktests.

(Dr. K. Sarnow)

Literatur

[1] Sarnow, K. Einführung in die künstliche Intelligenz. ST-Computer /87.

[2] Winston, P. H., B. K. P. Horn. LISP. Addison Wesley, 1984.

[3] Wirth, N. Algorithmen und Datenstrukturen. Teubner, 1983.


```
bedroht(I,J,A,B):-
     I =:= A;
     J =:= B;
     (I - J) =:= (A - B);
     (I + J) =:= (A + B).


konflikt(N,M,[]):-fail.
konflikt(N,M,[[A,B]|C]):-
     bedroht(N,M,A,B);
     konflikt(N,M,C).

queen(Size):- queen_aux([],0,Size).

queen_aux(Brett,N,N):-printr(Brett).
queen_aux(Brett,N,Size):-
     queen_sub(Brett,N,0,Size).

queen_sub(Brett,N,M,M).
queen_sub(Brett,N,M,Size):-
    (      konflikt(N,M,Brett)
     ;
           (   J is N+1,
               queen_aux([[N,M]|Brett],J,Size)
           )
    ),
     K is M+1,
     queen_sub(Brett,N,K,Size).
```

Listing 2: Der Benchmarktest in PROLOG



Klappe auf zur Vierten!

Willkommen zur Juni-Ausgabe der St-Ecke. Viele öffnen gerade bei diesem schönen Wetter ihre Fenster. Und genau dies wollen auch wir tun: nach mehrfacher Anfrage von Lesern soll die Routine **do_redraw()**, die in jedem zweiten Programm vorkommt und wichtig für die Fensterverwaltung ist, genauer unter die Lupe genommen werden, damit auch auf unserem ST die Fenster geöffnet werden können. Als zweites Thema wird die **variable Gestaltung von Menüleisten** behandelt. Den interessantesten Punkt der heutigen ST-ECKE bietet sicherlich die **Pexec**-Routine: Ich werde genauer auf die Optionen dieser Routine eingehen; zum Beispiel, wie man Programme lädt und zu einem späteren Zeitpunkt startet. Zum Schluß bieten wir eine **Joystick-Routine in OMIKRON-Basic** an, das sich in letzter Zeit immer größerer Beliebtheit erfreut.

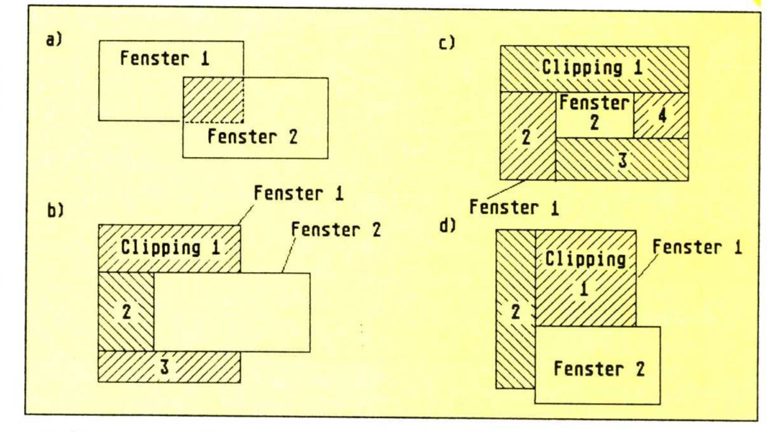


Abbildung 1: Möglichkeiten der Clippingbereiche

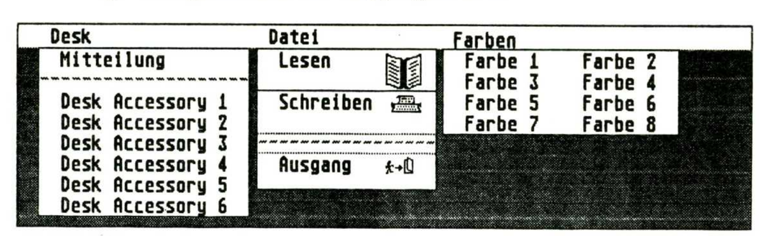


Abbildung 2: Menüzeile, einmal anders

Öffnet die Fenster

Zunächst einige Erklärung zur Fensterverwaltung des ST's. Vielfach versteht man unter einer Fensterverwaltung folgendes: Man definiert sich einen Bereich, in dem man mit seinem Fenster arbeiten möchte, und sieht die linke obere Ecke als XY-Koordinate 0,0 an. Außerdem erwartet man, daß, wenn der Fensterbereich verlassen wird, ein Überschreiben des Außenbereichs verhindert wird. Diese beiden Dinge (Koordinate 0,0 und Abfrage des Außenbereichs) werden leider vom ST nicht direkt übernommen. Allerdings bietet der ST Hilfen zur Handhabung des Fensterrahmens und dessen Objekte, wie zum Beispiel die Slider (Schieber), Fensterüberschrift etc., und gibt eine Meldung aus, wenn in ein Fenster überschrieben wurde, so daß das Programm, das das entsprechende Fenster bearbeitet, dieses Fenster wieder in Ordnung bringen kann. Das bedeutet also, daß der Inhalt des Fensters erstens immer wieder der entsprechende Fensterkoordinate angepaßt werden muß, falls das Fenster verschoben wurde, da die Koordinaten, wie eben erwähnt, absolute und *nicht* relative Koordinaten sind, und die Fensterinhalte bei Änderung des Bildschirms *teilweise* neu gezeichnet werden müssen.

Zur Erklärung des Wortes 'teilweise' schauen Sie sich bitte Bild 1 a an: Sie erkennen, daß das Fenster 1 von Fenster 2 stückweise überlappt wird. Das bedeutet, daß Fenster 1 rundherum und nicht vollständig gezeichnet werden muß. Dazu zeichnet man den Inhalt des Fensters mehrfach, aber immer nur auf einen bestimmten Bereich beschränkt. Diese Bereiche bezeichnet man als Clipping-Bereiche. Setzt man auf einen bestimmten Bereich ein Clipping, so werden alle Ausgaben (wie Linien, Kreise, Text etc.) auf diesen Bereich beschränkt. In Bild 1 sieht man Möglichkeiten wie Clipping-Bereiche eingeteilt sein können – es sei hier vorweggenommen, daß die Einteilung der Clipping-Bereiche dem GEM obliegt und dem zum Fenster gehörenden Programm mitteilt. Die Koordinaten der Clipping-Bereiche, die in X-, Y-Koordinate, Breite und Höhe übermittelt werden, können über die GEM-Routine wind_get() erfahren werden. Im Höchstfall kann es vorkommen, daß vier Bereiche gezeichnet werden müssen. Dies geschieht in dem Fall, wenn ein Fenster wie in Bild 1 c überdeckt wird. Die Vorgehensweise: Nachdem das Programm eine REDRAW-Botschaft von GEM erhalten hat, reagiert das Programm damit, daß es sich die Clipping-Bereiche der einzelnen Fenster geben läßt und entsprechend der Anzahl der Clipping-Bereiche den Inhalt des Fensters zeichnet. Sind also drei Clipping-Bereiche vorhanden, so wird der Inhalt des Fensters dreimal gezeichnet – jedesmal mit einem anderen Clipping-Bereich. Zum Setzen des Clippingbereichs stellt GEM die Routine vs_clip() zur Verfügung. Wichtig ist zu wissen, daß die Definierung des Bereiches durch die Koordinaten der linken

oberen und der rechten unteren Ecke und nicht durch Angabe der linken oberen Ecke mit Höhe und Breite gegeben ist. Da aber das GEM den Clipping-Bereich in letzter genannter Form übermittelt, müssen wir die Koordinaten erst umrechnen. Die Umrechnung erfolgt in der Routine set_clip() und wird daraufhin GEM in vs_clip() mitgeteilt.

Die Vorgehensweise soll nun am Quellcode von do_redraw() (Listing 1) erläutert werden. Die Übergabeparameter sind die Koordinaten des Fensters, die man über wind_get (wi_handle, WF_WORKXYWH,.....) übermittelt bekommt. Zunächst wird die Maus ausgeschaltet, da sie beim Zeichnen stören würde. Der nächste Schritt ist, dem GEM mitzuteilen, daß man den Bildschirm neu aufbauen möchte. Hiermit schaltet man die Funktion von GEM aus, die dem Programm mitteilt, daß der Bildschirm neu gezeichnet werden muß. Würde man diesen Befehl nicht verwenden, so sähe GEM das Neuaufbauen des Bildschirm wiederum als eine Verletzung des Bildschirminhaltes an, und...das Programm bekäme eine erneute REDRAW-Meldung. Als nächstes werden die Fensterkoordinaten in die zweite GRECT-Struktur (siehe ST_ECKE Mai) geschrieben, um sie für den späteren Vergleich in zusammengefaßter Form vorliegen zu haben. Nun muß das Programm die Clipping-Bereiche einlesen: (Die Vorgehensweise ist ähnlich wie in GEMDOS beim Einlesen der Directory, bei dem man zunächst den ersten Eintrag mit einer Funktion (FSFIRST) einliest und dann mit einer anderen (FSNEXT) die folgenden. Synonym dazu gibt es unter GEM die Funktion wind_get (wi_handle, WF_FIRST XYWH...) und wind_get (wi_handle, WF_NEXTXYWH...). In der 12. Zeile wird demnach der erste Clipping-Bereich eingelesen. Das Ende einer Clipping-Bereichsliste erkennt man daran, daß die Breite und Höhe eines Clipping-Bereiches immer ungleich Null sein müssen. Das bedeutet, daß man das Suchen von folgenden Bereichen in dem Moment abbrechen kann, wenn ein Bereich keine Breite und Höhe mehr hat, was auch in Zeile 13 geschieht. In der folgenden Zeile wird die schon im Mai (ST-Ecke) erklärte Funktion benutzt, um die Schnittmenge zwischen Fenster und Clippingbereich zu errechnen. Die Funktion gibt TRUE zurück, falls eine Überschneidung vorhanden ist und der Schnittbereich ist in der zweiten GRECT-Struktur vorhanden. Diesen Schnittbereich übergibt man der schon erwähnten set_clip()-Routine und man zeichnet seinen Fensterinhalt. Danach holt man sich den nächsten Clipping-Bereich. Diese Vorgehensweise wird solange durchgeführt, bis alle Clipping-Bereiche abgearbeitet worden sind. Danach wird der 'Update' wieder eingeschaltet, um wieder REDRAW-Meldung von GEM zuzulassen, und die Maus aktiviert.

Soviel zur Fensterverwaltung in diesem Monat. Schließen wir unsere Fenster und wenden wir uns dem Menü zu – um genauer zu sein, wir wollen unsere Menüs (Menüleisten) etwas attraktiver gestalten.

Ein interessanteres Menü

Viele werden sich schon gewundert haben, daß das RCS (Es wird hier auf das RCS Version 1.4 Bezug genommen) beim Gestalten von Menübäumen nur vier Objektarten (TITLE, ENTRY, --- und BOX) für das Konstruieren von Menüleisten zur Verfügung stellt. Es ist aber möglich alle (!) Objektarten in Menüs unterzubringen. Was halten Sie davon, Icons in Menüs zu verwenden oder in einem DROP-DOWN-Menü zwei Einträge nebeneinander zu haben? Die Vorgehensweise ist eigentlich recht einfach; verwendet wird dabei das Clippboard des RCS.

Als Beispiel wollen wir uns mit der Einbindung von Icons in eine Menüleiste befassen. Zunächst öffnen Sie eine Dialogbox und postieren ein Icon in ihr. Dann laden Sie die Datenmenge eines Icons – Selektieren Sie dazu das Icon und wählen Sie im Menü den Punkt LOAD an. Danach nehmen Sie ihr Icon und schieben es auf das CLIPBOARD des RCS. Dieser Umweg in den Menübaum ist notwendig, da man Icons in einem Menübaum nicht laden kann! Nun machen Sie ihren Menübaum auf und fügen einen Eintrag in ein Drop-Down-Menü ein. Diesen Eintrag ziehen Sie so groß, daß das Icon hineinpaßt. Dann können Sie es in diesen Menüeintrag einfügen. Wichtig dabei ist, daß das Icon keine Berührung mit dem Rand des Eintrages hat, da bei Anwählen des Menüpunktes immer erst der Eintrag selbst angewählt werden soll und nicht das Icon, sonst wäre der Effekt, daß zwar das Icon, aber nicht der Eintrag selbst selektiert ist. Ein Beispiel für die Gestaltung einer Menüzeile mit Icons finden Sie in Bild 2. Um mehrere (auch andere) Objekte in einer Menüzeile unterzubringen, müssen Sie, genauso wie oben beschrieben, das Objekt über den Umweg des Clippboards in die Menüzeile bringen. Natürlich können Sie es dort beliebig oft kopieren. Auf diese Weise ist das rechte Drop-Down-Menü in Bild 2 entstanden. Diese Art der Menügestaltung ist dann zu empfehlen, wenn man viele Objekte in einem Drop-Down-Menü unterbringen muß, aber das Menü dadurch zu lang werden würde. Theoretisch ist es sogar möglich, selbstdefinierte Objekte in einer Menüzeile unterzubringen.

Eine kleine Anmerkung: Allgemein sollte man sich – so weit es geht – an die Standards der Menügestaltung halten, da eine 'verspielte' Menügestaltung eher von ihrem eigentlichen Sinn, ein einfaches und schnelles Anwählen von Befehlen, ablenkt. Mehr zu Standards der Menügestaltung und Möglichkeiten des Resource-Construction-Sets finden sich in unserem demnächst erscheinenden Sonderheft.

Das Ausführen von Tochterprozessen

Nun möchte ich mich ein wenig näher mit der Pexec-Routine beschäftigen. Über diese Routine des GEMDOS wurde schon mehrere Male geschrieben, aber leider wurde in den meisten Fällen nur auf die Option 0 (LOAD and GO) eingegangen. Die anderen Optionen wurden verschwiegen, wobei aber gerade Optionen 3 und 4 sehr interessant erscheinen, was ich auch an den vielen Leserzuschriften erkennen konnte.

Die Funktion Pexec des Gemdos ist eine Prozedur, die es ermöglicht, aus Ihrem eigenen Programm andere Programme aufzurufen. Dabei bietet der Aufruf ret = Pexec (modus, pfad, base, umg) durch Änderung des Modus folgende Möglichkeiten:

Modus	Bedeutung
0	sogenannter LOAD 'n' GO-Modus. Das aufgerufene Programm wird geladen und gleich gestartet.
3	Das Programm wird nur in den Speicher geladen.
4	Das im Speicher vorhandene Programm wird gestartet.

5 Die zu jedem Programm gehörige BASEPAGE wird angelegt, wobei Adressen der Daten- und Textsegmente nicht gesetzt werden.

In den nächstfolgenden Absatz nehme ich auf Listing 2 Bezug, das die Vorgehensweise beim Laden und gleichzeitigen Starten eines Programmes darlegt. Wie Sie aus der Parameterliste der Pexec-Routine entnehmen können, muß dazu der Modus 0 eingeschaltet werden. Als zweiten Parameter übergeben Sie nun den Pfad des Programmes, das geladen und gestartet werden soll. In unserem Programmbeispiel ist dies „programm.prg". Als dritten Parameter muß man in diesem Modus die Kommandozeile übergeben. Die Kommandozeile entspricht den bei als TTP angemeldeten Programmen übergebenen Parametern. So ist es möglich, dem Tochterprozess beliebige Information mitzugeben. Wichtig dabei ist, daß der eigentliche Text erst bei String-Position 1 anfängt und nicht bei 0. Die Begründung ist darin zu suchen, daß in String-Position 0 die Länge des Strings zu finden ist. Daraus ergibt sich natürlich automatisch, daß ein Strings nicht länger als 255 Zeichen sein darf! Im Programmbeispiel werden nun der Text und die Länge in den Kommandostring kopiert und mit als Parameter an die Pexec-Routine übergeben. Der letzte Parameter enthält einen Zeiger auf die sogenannte Umgebung des Programms. Dort können zum Beispiel Suchpfade angegeben sein, die dem Tochterprozess mit übergeben werden sollen. Die Anwendung bleibt ganz allein dem Tochterprozess überlassen, der die Strings bei Bedarf verwenden kann. Nach Aufruf der Pexec-Routine wird der Tochterprozess vollständig abgearbeitet. Wird dieser beendet, so wird die Kontrolle wieder an den Start-Prozess, in diesem Fall unser Programm, zurückgegeben. Wichtig dabei sind folgende Dinge: Erstens kann der Tochterprozess wiederum Tochterprozesse ausführen. Als Beispiel möchte ich hier das DESKTOP anführen. Dieses ruft zum Beispiel als Tochterprozess eine SHELL auf, die wiederum als Tochterprozess einen Compiler aufruft usw. Zweitens muß leider daraufhingewiesen werden, daß der Speicherbereich des Programms nach dem Beenden des Tochterprozesses wieder freigegeben wird. Der Nachteil wird uns bei den folgenden beiden Modi bewußt.

```
#include <osbind.h>

char *pfad,          /* Pfadname des zu ladenden Programms */
     *base,          /* Basepage - Programmdeskriptor      */
     *umg;           /* Umgebung (Environment)             */

char  kommando[257];                 /* Kommandostring */

/**************************************************************/
/*                                                            */
/* (long)ret = Pexec(int modus,char *pfad,char *base,char *umg) */
/*                                                            */
/**************************************************************/

main()
{
     int ret1;                  /* Rückgabewert des Tochterprozesses */

     appl_init();                                  /* wegen form_alert */

     pfad = "programm.prg";      /* Pfadname des zu ladenden Programmes */

     strcpy(&kommando[1],"Übergabe");          /* Übergabestring kopieren */
     kommando[0] = (char) strlen("Übergabe");              /* Länge setzen */

     base = (char*) Pexec(3,pfad,kommando,0l);          /* Programm LADEN */

     if ( base < 0l )              /* Fehler beim Laden aufgetreten  ? */
     {
          form_alert(1,"[1][Leider ist ein|Fehler aufgetreten.][ Abbruch ]");
          appl_exit();
          Pterm0();                                   /* Programmabbruch */
     }

     ret1 = Pexec(4,0l,base,0l);                   /* Programm ausführen */

     printf("Rückgabewerte des Prozesses: %d",ret1);
     Cnecin();                                /* Auf Tastendruck warten */

     appl_exit();
}
```

Listing 1

```
#include <osbind.h>

char *pfad,          /* Pfadname des zu ladenden Programms */
     *base,          /* Basepage - Programmdeskriptor      */
     *umg;           /* Umgebung (Environment)             */

char  kommando[257];                 /* Kommandostring */

/**************************************************************/
/*                                                            */
/* (long)ret = Pexec(int modus,char *pfad,char *base,char *umg) */
/*                                                            */
/**************************************************************/

main()
{
     int ret1;                  /* Rückgabewert des Tochterprozesses */

     appl_init();                                  /* wegen form_alert */

     pfad = "programm.prg";      /* Pfadname des zu ladenden Programmes */

     strcpy(&kommando[1],"Übergabe");          /* Übergabestring kopieren */
     kommando[0] = (char) strlen("Übergabe");              /* Länge setzen */

     base = (char*) Pexec(0,pfad,kommando,0l);     /* Laden UND Starten */

     if ( base < 0l )              /* Fehler beim Laden aufgetreten  ? */
     {
          form_alert(1,"[1][Leider ist ein|Fehler aufgetreten.][ Abbruch ]");
          appl_exit();
          Pterm0();                                   /* Programmabbruch */
     }

     printf("Rückgabewerte des Prozesses: %d",ret1);
     Cnecin();                                /* Auf Tastendruck warten */

     appl_exit();
}
```

Listing 2

```
100 ' Joystick Abfrage
110 ' H. Schneider
120 ' OMIKRON Basic
130 '
140 REPEAT
150    Joy(Nr,Xy)'                              Aufruf
160    IF Nr=255 THEN '                         Joystick 1?
170       IF Xy=128 THEN PRINT TAB (40);"Bumm" ENDIF
180       PRINT "Joystick 1",Xy
190    ENDIF
200    '
210    IF Nr=254 THEN '                         Joystick 2?
220       PRINT "Joystick 0",Xy
230       IF Xy=128 THEN PRINT TAB (40);"Krach" ENDIF
240    ENDIF
250    ' IF Nr<254 THEN PRINT TAB (40);Nr: ENDIF
260 UNTIL Nr=57'                                Bei "SPACE" soll er aufhören
270 Maus_On'                                    Maus REAKTIVIEREN, sonst Essig
280 MOUSEON '                                   Zur Kontrolle Maus anzeigen
290 END
300
310 DEF PROC Joy(R Nr,R Xy)'                    Die Werte zurückgeben
320    OPEN "K",1'                              Tastaturprozessor "öffnen"
330       PRINT #1, CHR$($14);'                 automatische Joystickmeldung
340       '
350       '             Es werden zwei Bytes gesendet
360       '                1 Byte Joystick Nummer
370       '                   254 Joystick 0  (Maus)
380       '                   255 Joystick 1
390       '                andere Codes entsprechen dem
400       '                SCAN-Code der Tastatur
410       '
420       '
430       '              Die gelieferten Werte
440       '              ---------------------
450       '
460       '                       oben
470       '
480       '                5       1       9
490       '
500       '       links    4       0       8    rechts
510       '
520       '                6       2      10
530       '
540       '                       unten
550       '
560       '
570       Nr= ASC( INPUT$(1))'   Byte 1
580       Xy= ASC( INPUT$(1))'   Byte 2
590    CLOSE 1
600 RETURN
610
620 DEF PROC Maus_On
630    OPEN "K",1'                      Tastaturprozessor 'öffnen'
640    PRINT #1, CHR$($15); CHR$(8);'   Joystickabfrage beenden + Maus ein
650    CLOSE 1
660 RETURN
```

Listing 3

Das Laden und Starten von Tochterprozessen

Interessant klingt das Laden und spätere Starten von Tochterprozessen, dem wir uns jetzt zuwenden wollen. Der Unterschied zu dem obigen Verfahren ist nur der, daß man sich den Rückgabeparameter der Pexec-Routine bei Modus 3 merkt. Er enthält einen Zeiger auf die sogenannte Basepage des Programms. Eine Basepage ist eine Struktur, die Zeiger auf Text- und Datensegment, deren Länge und noch einiges mehr enthält. Übergibt man nun diese Adresse der Pexec-Routine mit Modus 4, so kann man nachträglich dieses geladene Programm starten (siehe Listing 3). Hierbei sollte beachtet werden, daß bei Modus 4 der Pfadname natürlich nicht mehr berücksichtigt wird. Außerdem werden Kommandozeile und Umgebungsstrings bei Modus 3 übergeben und somit auch nicht bei Modus 4 berücksichtigt. Nun könnte man auf die interessante Idee kommen, mehrere Programme zu laden und je nach Bedarf aufzurufen, um eine Art von Accessories zu realisieren. Dies ist aber aus einem Grund leider so einfach nicht möglich: Jeder Prozessspeicherbereich wird nach Beendigung des Prozesses wieder freigegeben. Das bedeutet aber, das ein geladender Prozess nur einmal aufgerufen werden kann. Folglich ist es nicht möglich, eine einmal geladene Datenmenge mehrmals zu starten, sofern man nicht alle benötigten Daten vor dem Starten rettet. Sofern Sie an weiteren Informationen bezüglich Programm- und Speicherverwaltung des GEMDOS interessiert sind, möchte ich Sie ein zweites Mal auf das kommende Sonderheft aufmerksam machen, in dem sehr genau auf diese und andere Funktionen hingewiesen wird.

Dem Joystick auf die Kontakte gefühlt

In einer vorherigen Ausgabe der ST-Ecke wurde auf die Joystickabfrage näher eingegangen. Realisiert wurde sie damals für die Sprache c, und zwar mit Hilfe eines kleinen Inline-Assembler Programms. Will man diese Methode in Basic umsetzen, so kommen, zumindest für nicht Assembler Programmierer, einige Probleme auf. Aus diesem Grunde haben wir die Abfrage des 'Freudenknüppels' in reinem Basic, genauer gesagt in OMIKRON-Basic, realisiert.

Das Prinzip ist recht einfach. Zuerst öffnet man einen Kanal zum Tastaturprozessor, was in OMIKRON Basic sehr einfach vonstatten geht, und schickt dem Tastaturprozessor einen Befehl, der ihm mitteilt, daß er nun automatisch die Joystickwerte senden soll. Anschließend liest man von gleichem Kanal zwei Bytes ein, in denen alle gewünschten Informationen enthalten sind. Das erste Byte gibt Aufschluß darüber, welches Joystick betätigt wurde. '255' entspricht dabei Joystick 1 und '254' Joystick 2. Enthält das erste Byte einen anderen Wert, so wurde die Tastatur bestätigt. Der Wert entspricht dann dem SCAN-Code, d. h. es können alle Tasten abgefragt werden (z. B. die Funktionstasten, HELP und UNDO). Wenn also das erste Byte '255' oder '254' betrug, so enthält das zweite Byte direkt die Richtung des betreffenden Joystick. Die Werte der entsprechenden Stellungen sind im Listing angegeben.

Die gesamte Abfrage wurde in einer Prozedur realisiert, wobei die Werte aufgrund der sauber gestalteten OMIKRON-Prozeduren – an das Hauptprogramm zurückgegeben werden. Die Umsetzung auf ST- und GFA-Basic ist prinzipiell natürlich möglich. OMIKRON-Basic hat aber einen sehr entscheidenden Vorteil: Beim Experimentieren mit dem Tastaturprozessor kommt es nicht selten zu Systemabstürzen, und der Rechner muß normalerweise neu gebootet werden. Da OMIKRON-Basic auf Modul enthalten ist, ist sowohl der Interpreter **als auch das momentan bearbeitete Programm** nach einem Reset noch vorhanden. Spätestens nach dem zweiten Reset wird man dies zu schätzen wissen.

Viel Spaß mit dieser ST-Ecke wünscht

(SH) + (HS)

BUCH- UND SOFTWARE-BESTELLKARTE

Ich/Wir bestelle(n)

Menge	Bestell-Nr.	Autor/Titel	Stückpreis	gesamt DM
		GfA-BASIC-BUCH	**49,–**	
		Programm-Diskette zum GfA-Buch	**39,–**	

☐ Scheck über DM__________ ist beigefügt
☐ Per Nachnahme

Zuzüglich DM 5,– Versandkostenanteil.

Bitte beachten:
Es werden nur Festbestellungen berücksichtigt.
Eine Rückgabemöglichkeit besteht nicht.
Ausnahme nur bei Beschädigung.

Name

Vorname

Straße/Nr.

PLZ Ort

Datum

Unterschrift

ST-COMPUTER Kleinanzeigen-Auftrag

Bitte veröffentlichen Sie für mich folgende Kleinanzeige in der angekreuzten Rubrik

Biete an ☐ Hardware ☐ Software

Ich suche ☐ Hardware ☐ Software ☐ **Tausch** ☐ **Kontakte** ☐ Verschiedenes ☐

30 Buchstaben je Standardzeile – incl. Satzzeichen und Wortzwischenräume.
Groß- und Kleinbuchstaben verwenden, fettgedruckte Wörter unterstreichen.

Bearbeitung **nur gegen Vorausscheck** über den entsprechenden Betrag (keine Überweisung)

☐ privat = DM 7,– je Zeile incl. MwSt.
☐ gewerblich = DM 15,– je Zeile + MwSt.
☐ Chiffregebühr = DM 10,–

☐ Scheck über DM__________ ist beigefügt

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze.

Datum Unterschrift

Absenderangaben auf der Rückseite nicht vergessen

Kurzmitteilung

an die ST-Computer Redaktion

Zu dem Artikel __________________ in Heft____, Seite____
hätte ich folgendes zu bemerken:

☐ Ich möchte Ihnen folgendes Programm zur Veröffentlichung anbieten: (Kurzbeschreibung, Sprache, Länge in Druckerseiten, GEM/TOS)

☐ Ich kann über folgendes Thema berichten: (Tips & Tricks am ST, Hardware, Software, etc.)

☐ Ich möchte gerne Autor in der ST-Computer werden. Meine Fachgebiete: (z. B. LISP, Pearl, Modula-2, DBase, Assembler...)

☐ Ich möchte, daß folgendes Public-Domain Programm in Ihre Sammlung aufgenommen wird.

☐ Sonstiges

Bei weiteren Angaben oder Fragen wenden Sie sich bitte schriftlich oder telefonisch an die Redaktion. Tel. 06196/482158

ST-COMPUTER Abonnement

Ja, bitte senden Sie mir die ST-Computer Fachzeitschrift ab__________
für mindestens 1 Jahr (11 Hefte) zum ermäßigten Preis von jährlich DM 60,– frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 80,– Normalpost.)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name

Vorname

Straße/Nr.

PLZ Ort

Datum Unterschrift

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. BLZ

Institut Ort

☐ Ein Verrechnungsscheck über DM______ liegt bei.

☐ Gegen Rechnung

Garantie:
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum Unterschrift

Absender

Name

Vorname

Straße/Postfach

PLZ/Ort

Telefon:

Postkarte

Heim-Verlag
Heidelberger Landstraße 194

6100 Darmstadt-Eberstadt

Telefon 0 61 51 / 5 60 57

Absender

Name, Vorname

Straße

PLZ/Ort

Telefon

Postkarte

ST-Computer
Redaktion
Schwalbacherstr. 64

6236 Eschborn

ST - Kleinanzeigen

Name

Vorname

Firma

Straße/Postfach

PLZ Ort

Datum Unterschrift

Bei Jugendlichen unter 18 Jahren bitte Unterschrift des Erziehungsberechtigten.

Postkarte

Bitte mit 60 Pf. frankieren

Anzeigenservice
Heim-VERLAG
Heidelberger Landstraße 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51/5 60 57

ST ABO

Straße/Postfach

PLZ Ort

Telefon:

Postkarte

Bitte mit 60 Pf. frankieren

Heim-VERLAG
Heidelberger Landstraße 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51/5 60 57

Einkaufsführer

Hier finden Sie Ihren Atari Fachhändler

Anzeigenschluß Heft 7/86: 25. Mai 1987

1000 Berlin



Computare
Keithstr. 18-20 • 1000 Berlin 30
030/21 390 21
186 346 com d

DATAPLAY
Bundesallee 25 · 1000 Berlin 31
Telefon: 030/861 91 61

Digital-Computer
Knesebeckstr. 76 · 1000 Berlin 12
Telefon
030-8 82 77 91



1000 Berlin





ATARI
... wir machen Spitzentechnologie preiswert.
Vertragshändler
UNION ZEISS
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61

2000 Hamburg

Gerhard u. Bernd Waller GbR
Computer & Zubehör-Shop
Kieler Straße 623
2000 Hamburg 54
040/570 60 07 + 570 52 75

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/49 44 00
Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon Sa. Nr. 0 40/641 50 91

2000 Hamburg

Hardware
Software
Beratung
Service



HABA COMPUTER AG

ATARI Systemfachhändler
Münsterstraße 9 · 2000 Hamburg 54
Telefon 040/56 60 1-1

2000 Norderstedt



2120 Lüneburg

Sienknecht
Bürokommunikation
Beratung - Verkauf - Werkstatt
Heiligengeiststr. 20, 2120 Lüneburg
Tel. 0 41 31 / 4 61 22, Btx 40 24 22
Mo.-Fr. 9[00]-18[00] und Sa. 9[00]-13[00]

2160 Stade

BERGAU
Büromaschinen · EDV-Systeme
Neue Straße 5, 2160 Stade
Telefon: (04141) 23 64 + 23 84

2210 Itzehoe

Der Computerladen
Inhaber Ulrich Bübel Martin Kopplow
Coriansberg 2 · 2210 Itzehoe
Telefon (0 48 21) 33 90/91

2300 Kiel



Die Welt der Computer
Dreiecksplatz Nr. 7
2300 Kiel 1 · ☎ 04 31 / 56 70 42

2350 Neumünster



Klosterstraße 2 · 2350 Neumünster
Telefon (0 43 21) 4 39 33

2390 Flensburg

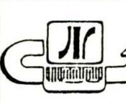


2800 Bremen

PS-DATA
Doventorsteinweg 41
2800 Bremen
Telefon 04 21 - 17 05 77

2850 Bremerhaven

HEIM- UND PERSONALCOMPUTER



Kurt Neumann
Georgstraße 71
2850 Bremerhaven
Tel. 04 71 / 4 20 06

HARDWARE · SOFTWARE · PAPIERWARE

ST-Computer Einkaufsführer

Werbewirksam, aktuell und preiswert.

Sprechen Sie mit uns:
Heim-Verlag
06151 / 56057

2940 Wilhelmshaven

Radio Tiemann GmbH & Co. KG
2940 Wilhelmshaven
Telefon 0 44 21/2 61 45

2950 Leer



- HARDWARE-SOFTWARE
- SYSTEM-ENTWICKLUNG
- ORGANISATION
- EDV-SCHULUNG
- EDV-BERATUNG
- SERVICE-WARTUNG

Augustenstraße 3 · 2950 Leer
Telefon 04 91 - 45 89

3000 Hannover

Pro-Computer Hannover
Inh. HELGA PROSCHEK
● **Beratung** ● **Verkauf**
● **Programmierung** ● **Installation** ● **Service**
☎ 05 11 / 52 25 79
D-3000 Hannover 71 · Großer Hillen 6

DATALOGIC COMPUTERSYSTEME
ATARI ST- COMPUTER HARDWARE SOFTWARE
BERATUNG SERVICE VERKAUF
CALENBERGER STR. 26
3000 HANNOVER 1
TEL. 0511 - 32 64 89

COM DATA
Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36

trend DATA Computer
IBM · EPSON · TRIUMPH ADLER
HEWLETT PACKARD · ATARI etc.
trendDATA Computer GmbH
Am Marstall 18-22 · 3000 Hannover 1
Telefon (05 11) 1 66 05-0

3040 Soltau

F & T Computervertrieb
Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 0 51 91 / 1 65 22

3100 Celle

Ludwig Haupt jr.
Büro-Einkaufs-Zentrum
Gerhard-Kamm-Straße 2
Ruf 8 30 45, Postfach 140
3100 Celle

3150 Peine

Wieckenberg & Schrage GmbH
Computertechnik
Hard- u. Software
Woltorfer Str. 8, 3150 Peine
Tel. 0 51 71 / 60 52/3 o. 0 51 73 / 79 09

3170 Gifhorn

COMPUTER-HAUS GIFHORN
Braunschweigerstr. 50
3170 Gifhorn
Telefon 0 53 71 - 5 44 98

3300 Braunschweig



Rebenring 49-50
3300 Braunschweig
Tel. (05 31) 33 32 77 / 78

3400 Göttingen

Büroeinrichtungs-Zentrum
Wiederholdt
3400 Göttingen-Weende
Wagenstieg 14 – Tel. 05 51 / 3 40 31

3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler
Rudolf-Schwander-Str. 5, 9 + 13
3500 Kassel
Tel. (05 61) 70 00 00

3550 Marburg

L W M
COMPUTER SERVICE
Bahnhofstraße 26b
3550 Marburg/Lahn
✆ 0 64 21 - 6 22 36

4000 Düsseldorf

BERNSHAUS GmbH
Bürotechnik – Bürobedarf
Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

4000 Düsseldorf

HOCO
EDV ANLAGEN GMBH
Flügelstr. 47
4000 Düsseldorf
Tel. 02 11 - 77 62 70

4050 Mönchengladbach



Hindenburgstr. 249
4050 Mönchengladbach
Tel. 0 21 61 - 187 64

4200 Oberhausen

KAMP
Büro- und Computersysteme
Vestische Straße 89/91
4200 Oberhausen 12 (Osterfeld)
Fernruf (02 08) 89 00 86
Fernschreiber 8 56 578

Hier könnte
Ihre Anzeige
erscheinen.
Anruf genügt:
Heim-Verlag
✆ 06151-56057

4300 Essen

Computerservice
Keese
Haus-Berge-Straße 163 · 4300 Essen 11
☎ (02 01) 66 70 62

ATARI Systemfachhändler



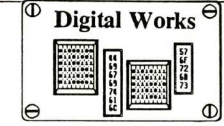
KARSTADT Aktiengesellschaft
Limbecker Platz 4300 Essen 1
Tel.: (02 01) 17 63 99

4400 Münster



COMPUTER SYSTEME GMBH
Daimlerweg 39 - 4400 Münster
Telefon 02 51 / 71 99 75 - 9

4410 Warendorf



Computer-Fachhandel – Hardware & Software
Jörg Kellert – Helmut Müller GbR
Brünebrede 17 · 4410 Warendorf
Tel. 0 25 81 / 6 11 26

Alle bisher erschienenen Ausgaben ST-Computer gibt's natürlich bei Ihrem Fachhändler

4422 Ahaus

ATARI · Epson · Fujitsu · Molecular · NCR · Tandon · Schneider · Star
OCB

OCB-Computershop	OCB-Hard- und Software
Wallstraße 3	Wessumerstraße 49
4422 Ahaus	4422 Ahaus
Tel. 0 25 61/50 21	Tel. 0 25 61/50 21

4430 Steinfurt

ATARI SCHNEIDER STAR NEC SEIKOSHA PANASONIC EPSON
Computer
Büromaschinen
Service
Telefon 02551/2555
Tecklenburger Str. 27 · 4430 Steinfurt

4500 Osnabrück

Heinicke-Electronic
Kommenderiestr. 120·4500 Osnabrück
Telefon 05 41 - 8 27 99
Wir liefern Micro-Computer seit 1978

4600 Dortmund

Bürostudio BOLZ
Brauhausstraße 4 · 4600 Dortmund
Telefon 02 31 - 52 77 13 - 16

Atari, Genie, Schneider, Tandy, Brother, Star, Memorex, BASF, Verbatim
cc Computer Studio GmbH
Software-Hardware-Beratung
Service-Eilversand
Ihre Ansprechpartner:
v. Schablinski
Jan P. Schneider
Elisabethstraße 5
4600 Dortmund 1
T. 0231/528184 · Tx 822631 cccsd

4600 Dortmund



City Elektronic
Güntherstraße 75
4600 Dortmund
Telefon 02 31/57 22 84

4650 Gelsenkirchen-Horst



Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel
Poststr. 15 · 4650 Gelsenkirchen-Horst
Tel. 0209/52572

4700 Hamm

computer center



4800 Bielefeld

hardware
software
organisation
service
CSF
CSF COMPUTER & SOFTWARE GMBH
Heeper Straße 106 – 108
4800 Bielefeld 1
Tel. (05 21) 6 16 63

Carl-Severing-Str. 190
4800 Bielefeld 14
Telefon: 05 21/45 99-150
Telex : 9 37 340 krab d
Telefax: 05 21/45 99-123
MICROTEC
Software
Hardware
Beratung
Service

5000 Köln

BÜRO MASCHINEN
braun
AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
RUF: 02 21 / 21 91 71

5010 Bergheim

Computerstudio HÖLSCHER

EDV-Beratung · Organisation
Programmierung · Home/Personal-Computer
Software · Zubehör · Fachliteratur

Zeppelinstr. 7 · 5010 Bergheim
Telefon 0 22 71 - 6 20 96

5060 Bergisch-Gladbach

Computer Center

Buchholzstraße 1
5060 Bergisch-Gladbach
Telefon 0 22 02 - 3 50 53

5090 Leverkusen

Rolf Rocke
Computer-Fachgeschäft
Auestraße 1
5090 Leverkusen 3
Telefon 0 21 71 / 26 24

5200 Siegburg

Computer Center

Luisenstraße 26
5200 Siegburg
Telefon 0 22 41/6 68 54

5400 Koblenz

SCHMITT COMPUTERSYSTEME

Casinostraße 40
5400 Koblenz
✆ 02 61 - 3 65 28

5412 Ransbach

Computer Technik Kieckbusch GmbH

Der Softwarespezialist

Am Seeufer 11 + 22 · 5412 Ransbach
Telefon 0 26 23 - 16 18

5457 Straßenhaus

DR. AUMANN GMBH
Computer-Systeme

Schulstr. 12
5457 Straßenhaus
Telefon 0 26 34 - 40 81/2

5500 Trier



Güterstr. 82 - 5500 Trier
☎ 06 51 - 2 50 44

Fordern Sie unsere Zubehör-Liste an.

5540 Prüm

ATC COMPUTER J. M. ZABELL

Ritzstraße 13 · Pf. 10 51
5540 PRÜM
– Tel.: 0 65 51 - 30 39 –

5600 Wuppertal

Jung am Wall

Wall 31—33
5600 Wuppertal 1
Telefon 02 02/45 03 30

MEGABYTE
Computer Vertriebs GmbH

Friedrich-Engels-Allee 162
5600 Wuppertal 2 (Barmen)
Telefon (02 02) 8 19 17

5630 Remscheid

C O M SOFT

Scheiderstr. 12 · 5630 Remscheid
Telefon (0 21 91) 2 10 33 - 34

5800 Hagen

ATARI
... wir machen Spitzentechnologie preiswert.

Vertragshändler **Axel Böckem**
Computer + Textsysteme

Elper Str. 60 (Eilpezentrum) · 5800 Hagen
Tel. 0 23 31 / 7 34 90

Hier könnte
Ihre Anzeige
erscheinen.

Anruf genügt:
Heim-Verlag
✆ 0 61 51 - 5 60 57

5900 Siegen

Hees Computer
Vertriebs GmbH
Hardware · Software · Schulung

Siegen · Weidenauer Str. 72 · ☎ 02 71/7 34 95

6000 Frankfurt

Müller & Nemecek

Kaiserstraße 44
6000 Frankfurt/M.
Tel. 0 69 - 23 25 44

GES-COMPUTER
GESELLSCHAFT FÜR EDV UND SOFTWARE mbH

Filiale Frankfurt	Filiale Hanau
Hartmann-Ibach-Str. 63	Steinheimer Str. 22
6000 Frankfurt 60	6450 Hanau
Tel.: (0 69) 46 20 41	Tel.: (0 61 81) 2 48 26

WAIZENEGGER
Büroeinrichtungen

Kaiserstraße 41
6000 Frankfurt/M.
☎ 0 69 / 2 73 06 - 0

ATARI
... wir machen Spitzentechnologie preiswert.

Vertragshändler
Jetzt bei uns.
bco
Büro-Computer + Organisations GmbH
Oederweg 7–9
6000 Frankfurt/M. 1
☎ (0 69) 55 04 56-57
Hardware ★ Software ★ Beratung ★ Service

SCHMITT COMPUTERSYSTEME

Grosse Friedbergerstr. 30
6000 Frankfurt
✆ 0 69 - 28 40 65

6100 Darmstadt

Heim
Büro- und Computermarkt
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
✆ 0 61 51 / 5 60 57

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Elisabethenstr. 15 · 6100 Darmstadt
Luisencenter · Tel. 0 61 51 - 10 94 20

6100 Darmstadt

SCHMITT COMPUTERSYSTEME

Mühlstraße 76
6100 Darmstadt
✆ 0 61 51 - 2 45 74

6200 Wiesbaden

SCHMITT COMPUTERSYSTEME

Rheinstraße 41
6200 Wiesbaden
✆ 0 61 21 - 30 73 30

6240 Königstein

KFC COMPUTERSYSTEME

Wiesenstraße 18
6240 Königstein
Tel. 0 61 74 - 30 33
Mail-Box 0 61 74 - 53 55

6300 Gießen



Schneider ATARI Commodore

BAUMS

BÜRO · ORGANISATION
Bahnhofstr. 26 · 6300 Gießen
Telefon: 06 41 / 7 10 96

6330 Wetzlar



Fachmarkt für Computer u. Unterhaltungs-electronic in Wetzlar,
Einkaufszentrum Bahnhofstraße, Tel. (0 64 41) 4 85 66

6400 Fulda

Schneider ATARI Commodore

WEINRICH

BÜRO · ORGANISATION
Ronsbachstraße 32 · 6400 Fulda
Telefon: 06 61 / 4 92-0

6457 Maintal

Landolt-Computer

Beratung · Service · Verkauf · Leasing

Wingertstr. 112
6457 Maintal/Dörnigheim
Telefon 0 61 81 - 4 52 93

6500 Mainz

:ELPHOTEC

Computer Systeme

Ihr Atari Systemhändler mit eigenem Service-Center

Walpodenstraße 10
6500 Mainz
Telefon 0 61 31 - 23 19 47

SCHMITT COMPUTERSYSTEME

Karmeliterplatz 4
6500 Mainz
✆ 0 61 31 - 23 42 23

6520 Worms

ORION

Computersysteme GmbH
Friedrichstraße 22
6 5 2 0 W O R M S
Tel. 0 62 41 / 67 57 - 67 58

6700 Ludwigshafen

MKV Computermarkt

Bismarck-Zentrum
6700 Ludwigshafen
Telefon 06 21 - 52 55 96

6720 Speyer

MKV Computermarkt

Gilgenstraße 4
6720 Speyer
Telefon 0 62 32 - 7 72 16

6750 Kaiserslautern

C.O.S.

COMPUTER ORG. GmbH

Karl-Marx-Straße 8
6750 Kaiserslautern
Telefon (06 31) 6 50 61 - 62

6750 Kaiserslautern



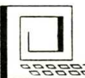
6800 Mannheim

GAUCH + STURM

Computersysteme + Textsysteme

6800 Mannheim 24

Casterfeldstraße 74-76
☎ (06 21) 85 00 40 · Teletex 6 211 912



Computer-Center
am Hauptbahnhof GmbH

L 14, 16-17
6800 Mannheim 1
Tel. (06 21) 2 09 83 / 84

6900 Heidelberg

JACOM COMPUTERWELT

Hardware · Software
Schulung · Service

Mönchhofstraße 3 · 6900 Heidelberg
Telefon 0 62 21 / 41 05 14 - 550

Heidelberger Computer-Center

Bahnhofstraße 1
6900 Heidelberg
Telefon 0 62 21 / 2 71 32

7000 Stuttgart



COMPUTERFACHHANDEL

Der Computer-spezialist

star · ATARI · olivetti · NEC

BNT Computerfachhandel GmbH Marktstr. 48 7000 Stuttgart 50
Tel.: Büro/Service (07 11) 55 83 83 Hotline (07 11) 55 83 92
Tel.: Verkauf (07 11) 55 83 91 Telefax (07 11) 56 70 93
Filiale: Karlstraße 12, 7140 Ludwigsburg, Telefon (07 141) 90901

SCHMITT COMPUTERSYSTEME

Tübingerstr. 18
7000 Stuttgart

7022 L-Echterdingen

Autorisierter ATARI-System-Fachhändler

ATARI ST

Matrai computer

Matrai Computer GmbH
Bernhauser Str. 8
7022 L.-Echterdingen
☎ (0711) 797049

7030 Böblingen

Verkauf – Service – Software



Norbert Hlawinka
Sindelfinger Allee 1
7030 Böblingen
Tel. 07031/226015

COMPUTER CENTER

7100 Heilbronn

Unser Wissen ist Ihr Vorteil

Walliser & Co.
Mönchseestraße 99
7100 Heilbronn
Telefon 07131/60048

Computer-Welt



Am Wollhaus 6
7100 Heilbronn
Tel. 07131-68401-02

7150 Backnang



7410 Reutlingen

Computer-Shop
Werner Brock
autorisierter Fachhändler f. ATARI, Schneider, Oki

Federnseestr. 17 · 7410 Reutlingen
Telefon: 07121/34287

7450 Hechingen



Gesellschaft für Datenverarbeitung mbh

Computer · Drucker
Zubehör · Fachliteratur

Schloßplatz 3 · 7450 Hechingen
Telefon 07471/14507

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis

HEIM + PC-COMPUTERMARKT

HARDWARE · SOFTWARE · LITERATUR

SCHEURER

ATARI COMMODORE CUMANA DATA-BECKER
MULTITECH RITEMAN SCHNEIDER THOMSON

7475 Meßstetten 1 · Hauptstraße 10 · 07431/61280

7480 Sigmaringen



7500 Karlsruhe

papierhaus erhardt

Am Ludwigsplatz · 7500 Karlsruhe
Tel. 0721-23925

MKV Computermarkt

Rüppurer Straße 2d
7500 Karlsruhe
Telefon 0721-373071

7530 Pforzheim

DM Computer GmbH
Hard- & Software
Durlacherstr. 39
Tel. 07231-13939
7530 Pforzheim
Telex 783248

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk

In der Jeuch 3
7600 Offenburg
Telefon 0781/57974

7640 Kehl/Rhein



eigener Service · eigene Software

Badstrasse 12
Tel. 07807/822
Telex: 752913
7607 NEURIED 2

Filiale:
Hauptstrasse 44
Tel. 07851/1822
7640 KEHL/RHEIN

ELEKTRO-MÜNTZER GmbH

7700 Singen

U. MEIER
Computersysteme

7700 Singen-Htwl.
Am Posthalterswäldle 8
Telefon 07731-44211

7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK

Villinger Straße 85
7730 VS-Schwenningen
Telefon 07720/38071-72

7750 Konstanz

ATARI ★ PC's ★ SCHNEIDER

computer - fachgeschäft
rösler
Rheingutstr. 1 · ☎ 07531-21832

7800 Freiburg

CDS
EDV-Service GmbH

Windausstraße 2
7800 Freiburg
Tel. 0761-81047



7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 07751/3094

7900 Ulm

HARD AND SOFT COMPUTER GMBH
Ulms großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Herrenkellergasse 16 · 7900 Ulm/Donau
Telefon 0731/62699

7900 Ulm

COMPUTERSTUDIO
Büro & Datentechnik · 2x in Ulm
Claus Wecker
Hafenbad 18/1 + Frauenstr. 28
7900 Ulm/Do.
Telefon (07 31) 2 80 76

7950 Biberach

HARD AND SOFT COMPUTER GMBH
Biberachs großes Fachgeschäft für BTX, Heim- u. Personalcomputer
Schulstraße 6 · Bei der VHF
7950 Biberach · Tel. 0 73 51 / 1 22 21

7980 Ravensburg

GRAHLE
Expert Grahle Computer
Eisenbahnstr. 33
7980 Ravensburg
Tel.: 0751/15955
Vertragshändler für ATARI, Schneider und Star

8000 München



COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STR. 62 L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089/3113066 · TELETEX 898341

SCHMITT COMPUTERSYSTEME
Arnulfstraße
8000 München





ATARI COMPAQ
Das Computer-Fachgeschäft im HERTIE · Hauptbahnhof
Bahnhofsplatz 7, 8000 München 2
Telefon 089/595277
Ein Unternehmen der procomp -Gruppe

Hier könnte
Ihre Anzeige
erscheinen.
Anruf genügt:
Heim-Verlag
✆ 06151 - 56057

8000 München

schulz computer
Schillerstraße 22
8000 München 2
Telefon (0 89) 59 73 39
Beratung · Verkauf · Kundendienst

8032 Gräfeling



COMPUTER SYSTEME SCHULUNG
Am Haag 5
8032 Gräfelfing
Tel. 089-8545464,851043

8070 Ingolstadt

DREYER GMBH
Elektrotechnik
Manchinger Straße 125
8070 Ingolstadt
Tel. 08 41 / 65 90

S & S
Computer-Center GmbH
Kuperstraße 20 · 8070 Ingolstadt
Tel. 08 41 - 3 28 12

8120 Weilheim

Klement
Elektro-, Radio- und Fernseh-Center
Beleuchtungskörper · Schallplatten
Meisterbetrieb · Computer Fachhändler
8120 Weilheim · Admiral-Hipper-Straße 1
Geschäft ✆ 45 00 · Kundendienst ✆ 44 00
Interfunk-Fachgeschäft

8150 Holzkirchen

ATARI
Besuchen Sie uns!
Fordern Sie unseren Software-Katalog (520ST) an!

M
MÜNZENLOHER GMBH
Tölzer Straße 5
D-8150 Holzkirchen
Telefon: (0 80 24) 18 14

8170 Bad Tölz

Langheinrich Elektronik
Bad Tölz
Wachterstraße 3
8170 Bad Tölz
Telefon 0 80 41 / 4 15 65

8220 Traunstein

computer studio
BÜROMASCHINEN cs
Ludwigstraße 3 — 8220 Traunstein
Stadtplatz 10 · Tel. 0861-14767 o. 3905

8330 Eggenfelden

Hot Space
Computer-Centrum
R. Lanfermann
Schellenbruckstraße 6
8330 Eggenfelden
Telefon 087 21/65 73
Altöttinger Straße 2
8265 Neuötting
Telefon 086 71/7 16 10

8400 Regensburg

C-SOFT GMBH
Programmentwicklung & Hardware
Holzfällerstraße 4
8400 Regensburg
Telefon 09 41 / 8 39 86

Zimmermann elektroland

8400 Regensburg	8390 Passau
Dr.-Gessler-Str. 8	Kohlbruck 2a
☎ 09 41 / 9 50 85	☎ 08 51 / 5 20 07

8490 Cham

A&P-SHOP®
STEUER



Hardware / Software
Computerzubehör
Büro- Schreibwaren
Papiergroßhandel
COPY - SHOP
A&P SOFTWARE
Auf der Schanze 4 · 8490 Cham/Opf. · Telefon (09971) 9723

8500 Nürnberg

EINE IDEE ANDERS
KARSTADT NÜRNBERG AN DER LORENZKIRCHE



1. KLASSE EINKAUFEN IM WELTSTADTHAUS

8500 Nürnberg

SCHMITT COMPUTERSYSTEME
Innere Laufer Gasse 29
8500 Nürnberg
✆ 09 11 - 20 97 17

8500 Nürnberg

ATARI

Der ATARI - Systemfachhändler in Nürnberg

Softwareberatung und Entwicklung

Computer • rucker • Zubehör • Fachliteratur



HIB Computer GmbH
Äußere Bayreuther Str. 72
8500 Nürnberg
T4el.:0911/515939

8520 Erlangen



Computerservice Decker

Meisenweg 29 - 8520 Erlangen
Telefon 09131 / 42076



Büro+Computer

8520 Erlangen
Dresdener Str. 5
Friedrichstr. 9
Tel.: 09131 / 12010

Alle bisher erschienenen Ausgaben ST-Computer gibt's natürlich bei Ihrem Fachhändler

8600 Bamberg



Bamberg · Tel. 0951/27808-09

8700 Würzburg

HALLER GMBH
Fachgeschäft für
Mikrocomputer
Büttnerstraße 29
8700 Würzburg
Tel. 0931/16705

8700 Würzburg

SCHOLL BÜROTEAM

Hardware · Software
Service · Schulung

computer center

am Dominikanerplatz
Ruf (0931) 50488

8720 Schweinfurt

Uhlenhuth GmbH

Computer + Unterhaltungselektronik

Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 09721 / 652154

8900 Augsburg

Adolf & Schmoll Computer

Schwalbenstr. 1 · 8900 Augsburg
Tel. (0821) 528533 oder 528087

Wir sind außerdem autorisierte Service-Fachwerkstatt für:

Schneider
commodore
ATARI

ATARI Systemfachhändler



KARSTADT Aktiengesellschaft
Bürgermeister-Fischer-Str. 6-10
8900 Augsburg · Tel. (0821) 3153-416

SCHMITT COMPUTERSYSTEME

Frauentorstr. 22
8900 Augsburg
✆ 0821-154268

8940 Memmingen

EDV-Organisation
Hard- + Software
Manfred Schweizer KG

Benninger Str. 34, Tel. 08331/12220

8940 Memmingen

Österreich

A-1020 Wien

W + H.
Computerhandel Ges m.b.H.

Förstergasse 6/3/2 · 1020 Wien
Tel. 0222-350968

A-1040 Wien

Ihr ST-Fachhändler in Wien

Computer-Studio

Wehsner Gesellschaft m.b.H.

1040 Wien - Paniglgasse 18-20
Telefon 0222-657808, 658893

A-8010 Graz



A-8010 Graz, Mandellstraße 23, Tel. (0316) 702840, 783923
Tlx.: 032534 zupan a

Schweiz

CH-1700 Fribourg

Softy Hard's Computershop

Die ATARI ST Spezialisten

Grand Rue 42
CH-1700 Fribourg
Tel. 0041(0) 37 22 26 28

CH-2503 Biel

UE
URWA Electronic

Ihr ATARI ST Spezialist in der Schweiz.
032 / 25 45 53

Lindenweg 24, 2503 Biel

CH-3012 Bern



Die ST-Spezialisten in Bern

Öffnungszeiten: Mo – Fr 9⁴⁵ – 18³⁰ – Sa 9⁰⁰ – 16⁰⁰

größtes Software- und Peripherieangebot der Region

Falkenplatz 7 · 3012 Bern · 061/244006

CH-3084 Bern

Computer Corner

S. Bazan
Seftigen Str. 240
3084 Bern-Wabern
Tel: 031-545100

Hardware
Software
Zubehör
Occasion

CH-3415 Hasle Rüegsau

COMPU-TRADE

Ihr ATARI Spezialist

Emmenstr. 16
CH-3415 Hasle-Rüegsau
✆ 0 34/61 45 93 auch abends bis 21.00 h

HARD- u. SOFTWARE · BERATUNG · EILVERSAND

CH-4054 Basel

COMPUTER CENTER

DIE ST-SPEZIALISTEN IN BASEL

Öffnungszeiten:

Di.–Fr.	9.30–12.30/14.00–18.30
Sa.	9.30–16.00
Mo.	geschlossen

SYSAG

COMPUTER SYSTEME · COMPUTER SCHULE

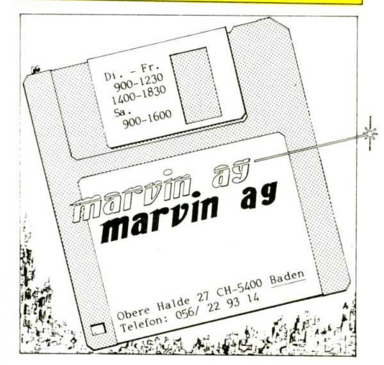
HOLEESTRASSE 87 · 4054 BASEL · TELEFON 061 39 25 25

CH-4625 Oberbuchsiten

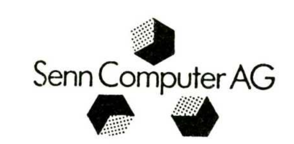
STECTRONIC M. Steck
Electronic-Computer-Shop

Hauptstr. 104/137
CH-4625 OBERBUCHSITEN
Tel. 0 62/63 17 27 + 63 10 27

CH-5400 Baden



CH-5430 Wettingen



Zentralstrasse 93
CH-5430 Wettingen

Tel. 056 / 27 16 60
Telex 814 193 seco

CH-8006 Zürich

ADAG Computershop

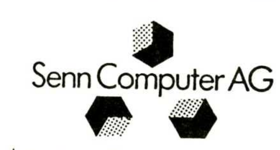
Universitätsstr. 25 · 8006 Zürich
Tel. 01 / 47 35 54

ATARI & WANG & EPSON

Computer-Center P. Fisch

Stampfenbachplatz 4
8006 ZÜRICH
✆ 01 / 363 67 67

CH-8021 Zürich



Langstrasse 31
Postfach
CH-8021 Zürich

Tel. 01 / 241 73 73
Telex 814 193 seco

VILAN

Das Warenhaus der neuen Ideen

Ihr Computer-Fachhandel an der
Bahnhofstrasse 75 · Zürich

CH-8050 Zürich



CH-8200 Schaffhausen

ZIMELEC

CAR-AUDIO
COMPUTER + ELECTRONIC
Bachstraße 28 · 8200 Schaffhausen
Tel.: 053/5 52 24

Montag—Freitag	9.00—12.00
	13.30—18.30
Samstag	9.00—16.00

CH-9000 St. Gallen

VIDEO - COMPUTER - CENTER
GÄCHTER AG

Webergasse 22
9000 St. Gallen
Telefon 071/ 22 60 05

video computer center

CH-9400 Rorschach

PAUS-electronic
Ihr Computer-Fachhandel

Industriestr. 30

CH-9400 Rorschach ✆ 071-41 18 85

CH-9450 Altstätten

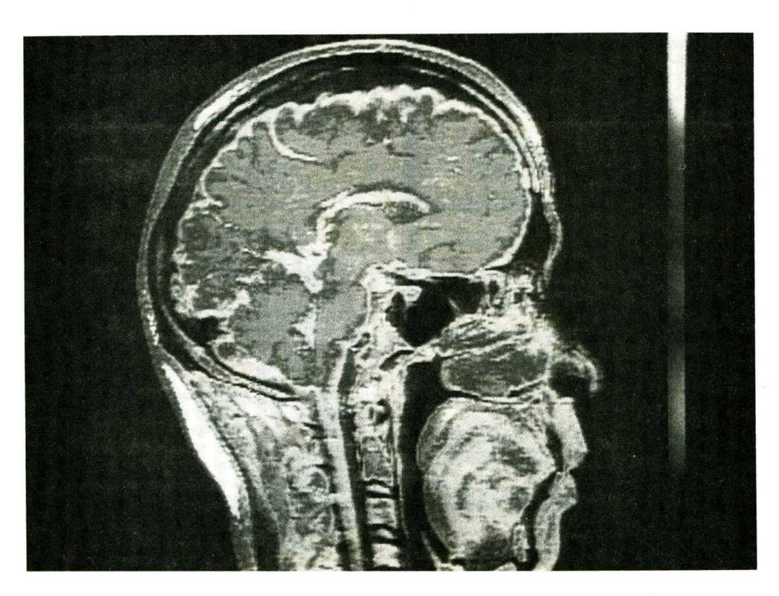
PAUS-electronic
Ihr Computer-Fachhandel

Ringgasse 27 · CH-9450 Altstätten
Telefon 0 71-75 34 79

Luxemburg







Elemente der künstlichen Intelligenz

Eine Einführung in die Programmiermethoden und Sprachen der KI

4. Teil: Automatisches Beweisen

Einführung in die Problematik

In der letzten Folge haben wir zusammen die ersten logischen Terms umgeformt und dabei gesehen, daß Sprachen wie PROLOG und LISP die logischen Termumformungen weitgehend übernehmen können. Viel interessanter ist natürlich das Problem, dem Computer die selbständige Führung eines mathematischen Beweises unter Benutzung logischer Termumformungen zu übertragen. Einem mathematisch wenig gebildeten Menschen bereitet es meist erhebliche Schwierigkeiten, einem mathematischen Beweis auch nur zu folgen. Einen mathematisch korrekten Beweisgang werden sich deshalb wohl auch die wenigsten Leser zutrauen. Sie werden sicher zustimmen, wenn man ein Gerät, das diese Beweise selbständig durchführt, als intelligent bezeichnet. Und da die Logik die „Haus- und Hofwissenschaft" der Informatiker ist, ist es nicht weiter verwunderlich, wenn auf dem Gebiet der automatischen Beweisführung Spektakuläres geleistet wurde.

Grundlagen der Aussagenlogik

Zunächst brauchen wir natürlich wieder einige Begriffe aus der Aussagenlogik. Ich darf bei dieser Gelegenheit wieder einmal auf das wirklich ausgezeichnete Buch von D. R. Hofstadter (siehe Literaturhinweise) hinweisen, in dem Sie eine sehr eigenwillige, gut lesbare Einführung in die Aussagenlogik finden. Generell ist es die Aufgabe eines Beweises, zu zeigen, daß eine wohlgeformte Aussage (d. h. eine Aussage, deren syntaktischer Aufbau den Regeln der Aussagenlogik genügt) immer den Wahrheitswert **wahr** erhält, egal welchen Wahrheitswert die einzelnen in der Aussage vorhandenen Variablen annehmen. Z. B. ist die Aussage **a v ~ a** immer wahr, egal welchen Wert **a** annimmt. Um zu beweisen, daß diese Aussage immer wahr ist, würde es genügen, wenn man die Wahrheitstafel der Aussage niederschreibt. Bei komplizierten Aussagen führt dieses Verfahren schnell an die Grenzen der Anschaulichkeit. Dann ist es einfacher, durch erlaubte Umformungen zum gewünschten Ergebnis zu kommen. Nehmen wir an, daß die Ausdrücke **a** und **b** wohlgeformt sind. Dann sind es auch alle mit Hilfe der Junktoren **~,&,v** gebildeten Ausdrücke: **~a, a v b, a & b.** Außerdem ist natürlich ein atomarer Ausdruck wohlgeformt. Weiterhin gelten die folgenden Äquivalenzen, d. h., der Ausdruck rechts von = kann ersetzt werden durch den Ausdruck links von = und umgekehrt:

1) ~ ~ a = a
   Gesetz der doppelten Verneinung
2) ~(a v b) = ~a & ~b
   1. De Morgan'sches Gesetz
3) ~(a & b) = ~a v ~b
   2. De Morgan'sches Gesetz
4) a v (b & c) = (a v b) & (a v c)
   Distributivgesetz
5) a → b = ~a v b
   Implikation

Die resultierenden Ausdrücke sind dann wieder wohlgeformt. Weiterhin benötigt man zur Herleitung neuer Aussagen noch sogenannte Schlußregeln:

1) Modus Ponens (bejahende Abtrennungsregel)

   A → b
   a
   ———
   b

   Beispiel: Wenn es regnet, dann ist der Rasen feucht.
   Es regnet.
   ———
   Der Rasen ist feucht.

2) Modus Tollens (verneinende Abtrennungsregel)

   a → b
   ~b
   ———
   ~a

Beispiel: Wenn es regnet, dann ist der Rasen feucht.
Der Rasen ist nicht feucht.

Es regnet nicht.

3) Kettenschluß

$$\frac{\begin{array}{c} a \rightarrow b \\ b \rightarrow c \end{array}}{a \rightarrow c}$$

Beispiel: Wenn Peter wütend ist, dann hat er Hunger.
Wenn Peter Hunger hat, dann ißt er Schokolade.

Wenn Peter wütend ist, dann ißt er Schokolade.

Die für uns interessante Form eines Beweises ist der sogenannte indirekte Beweis. Hier untersucht man die Negation des zu beweisenden Ausdrucks. Gelangt man durch Umformung der Negation zu einem Widerspruch, gilt also die negierte Aussage nicht, dann gilt nach dem Prinzip der doppelten Verneinung die originale Aussage. Schauen wir uns als Beispiel den Beweis des Kettenschlusses an:

Zunächst negieren wir den Term:

$\sim((a \rightarrow b) \,\&\, (b \rightarrow c) \rightarrow (a \rightarrow c)) =$
$\sim(\sim((\sim a \vee b) \,\&\, (\sim b \vee c)) \vee (\sim a \vee c))$

Diesen Term bringen wir in die konjunktive Normalform, d. h. wir formen ihn so um, daß er nur noch einzelne, durch & verbundene Terme enthält:

$\sim((\sim(\sim a \vee b) \vee \sim(\sim b \vee c)) \vee (\sim a \vee c)) =$
$(\sim a \vee b) \,\&\, (\sim b \vee c) \,\&\, \sim(\sim a \vee c) =$
$(\sim a \vee b) \,\&\, (\sim b \vee c) \,\&\, a \,\&\, \sim c$
(Term in konjunktiver Normalform.

Diese konjunktive Normalform eines logischen Termes heißt auch Klauselform, die einzelnen durch & verbundenen Terme Klauseln (also z. B. $(\sim a \vee b)$). Das Prinzip der Resolution besteht nun darin, zu zeigen, daß sich zwei Klauseln widersprechen, d. h. daß in einem logischen Term ein Unterterm der Form $\ldots \& \; x \; \& \sim x \; \& \ldots$ vorhanden ist. Das im folgenden Kapitel erläuterte Resolutionsprinzip geht von diesem leicht zu beweisenden Theorem aus:

$$(a \vee x) \,\&\, (\sim a \vee y) \rightarrow x \vee y$$
(Resolutionsschritt)

Damit würde sich der Term in konjunktiver Normalform darstellen lassen als:

$(\sim a \vee b) \,\&\, (\sim b \vee c) \,\&\, a \,\&\, \sim c = (a \vee 0) \,\&\, (\sim a \vee b) \,\&\, (\sim b \vee c) \,\&\, \sim c \rightarrow$

```
%----------------------------------------------------------------------------
% Interpreter für musterorientierte Programmierung
% nach I. Bratko, PROLOG Programming for artificial Intelligence.

:- op(800,xfx,--->).  % Definiert den "Ausführungsoperator"

start :-                        % Dieses Prädikat sucht in der
   Bedingung ---> Aktion,       % Datenbank  nach  einem  Muster
   test(Bedingung),             % [] ---> []. Alle Elemente der
   führe_aus(Aktion).           % ersten Liste werden als Bedingung
                                % getestet. Wenn alle Bedingungen
test([]).                       % erfüllt sind, werden alle Aktionen
test([Kopf|Rest]) :-            % der zweiten Liste ausgeführt.
   call(Kopf),
   test(Rest).

führe_aus([stop]):- told,abolish(klausel,1).
führe_aus([]):- start.
führe_aus([Kopf|Rest]) :-
   call(Kopf),
   führe_aus(Rest).

ersetze(A,B) :-
   retract(A),
   assert(B).


%----------------------------------------------------------------------------
% Übersetzung einer Aussage in Klauseln der Datenbasis

:- op(100,fy,~).     % Diese Direktiven definieren die logischen Operatoren
:- op(110,xfy,&).    % nicht (~), und (&), oder (v) und die Implikation (=>)
:- op(120,xfy,v).    % Das erste Argument ist die Priorität des Operators.
:- op(130,xfy,=>).   % Je kleiner die Zahl, desto größer die Priorität.

übersetze(F & G) :- % Dieses Prädikat übersetzt logische Terme in der
   übersetze(F),    % konjunktiven Normalform mit Hilfe der weiter unten
   übersetze(G).    % definierten erlaubten Umformungen. Ist keine weitere
                    % Umformung mehr möglich, wird der Term als Klausel
                    % abgespeichert.

übersetze(Aussage):-
   forme_um(Aussage,NeueAussage),write('('),write(Aussage),write(') => ('),
   write(NeueAussage),write(')'),nl,
   übersetze(NeueAussage).

übersetze(Aussage):-
   assert(klausel(Aussage)),write(klausel(Aussage)),nl.
forme_um(~(~X),X) .                           % Prinzip der doppelten Negation
forme_um(X => Y,~X v Y).                      % Die Implikation
forme_um(~(X & Y),~X v ~Y).                   % 1. De Morgansches Gesetz
forme_um(~(X v Y),~X & ~Y).                   % 2. De Morgansches Gesetz
forme_um(X & Y v Z,(X v Z) & (Y v Z)).        % 1. Distributivgesetz
forme_um(X v Y & Z,(X v Y) & (X v Z)).        % 2. Distributivgesetz
forme_um(X v Y,X1 v Y) :- forme_um(X,X1).     % Umformung von Teilausdrücken
forme_um(X v Y,X v Y1) :- forme_um(Y,Y1).     %    "        "         "
forme_um(~X,~X1) :- forme_um(X,X1).           %    "        "         "

%----------------------------------------------------------------------------
% Musterorientiertes Programm zur Durchführung der Resolution

% Widerspruch gefunden:

[klausel(X),klausel(~X)] --->
   [write('Widerspruch in der Negation gefunden!'),nl,
    write('Aussage damit wahr.'),nl,stop].

% (a v ~a v b) ist immer wahr.

[klausel(C),in(P,C),in(~P,C)] ---> [write(klausel(C)),  -
    write(' ist immer wahr, wird also aus der Datenbank entfernt.'),nl,
    retract(klausel(C))].

% (a v a v b) = (a v b) Idempotenzgesetz vereinfacht Terme.

[klausel(C),entferne(P,C,C1),in(P,C1)] --->
   [write(klausel(C)),
    ersetze(klausel(C),klausel(C1)),
    write(' wird ersetzt durch '),write(klausel(C1)),nl].

% Resolution Spezialfall: a & (~a v b) => b

[klausel(P),klausel(C),entferne(~P,C,C1),not fertig(P,C,P)] --->
```

Fortsetzung auf Seite 22



(0 v b) & (~b v c) & ~c = b & (~b v c) & ~c → <u>c & ~c</u>
(Widerspruch!)

Das Resolutionsprinzip

Das Resolutionsprinzip stellt sich also folgendermaßen dar.

1) Negiere den zu beweisenden Term.
2) Überführe den negierten Term in die Klauselform (konjunktive Normalform)
3) Führe wenn möglich den Resolutionsschritt durch.
4) Durchforste die Menge der Klauseln auf Widerspruch.
5) Wenn ein Widerspruch entdeckt wurde, dann ist die nicht negierte Aussage bewiesen.

Das in Listing 1 dargestellte PROLOG-Programm führt die hier beschriebene Resolution durch. Es benutzt einen in PROLOG geschriebenen Musterinterpreter aus dem sehr interessanten Buch von I. Bratko. Das Programm ist von mir so verändert worden, daß man jederzeit die Schlußweise der Maschine verfolgen kann. Gleichzeitig möchte ich an Hand dieses Programms die hervorragenden Eigenschaften von PROLOG bei der Definition eigener Operatoren erläutern.

Musterorientierte Programmierung

Unter musterorientierter Programmierung versteht man ein Programmschema, das nicht so starr reagiert wie die konventionellen Programmiermethoden. Normalerweise besteht ein Programm aus festen Modulen, deren Ablauf explizit vorgegeben ist. Ein musterorientiertes Programm besteht dagegen aus einer Sammlung von musterorientierten Modulen. Jedes Modul besteht aus zwei Teilen:

1) einem Bedienungsmuster,
2) Aktionen, die ausgeführt werden sollen, wenn das Bedingungsmuster erfüllt ist.

Die Ausführung eines Moduls wird in Gang gesetzt, wenn ein passendes Muster in der Datenbasis entdeckt wird. Musterorientierte Programmierung

Fortsetzung von Seite 21

```
    [assert(klausel(C1)),assert(fertig(P,C,P)),write('klausel('),write(P),
     write(') & klausel('),write(C),write(') => '),write(klausel(C1)),nl].

 % Resolution Spezialfall: ~a & (a v b) => b

 [klausel(~P),klausel(C),entferne(P,C,C1),not fertig(~P,C,P)] --->
    [assert(klausel(C1)),assert(fertig(~P,C,P)),write('klausel('),write(~P),
     write(') & klausel('),write(C),write(') => '),write(klausel(C1)),nl].

 % Resolution allgemeiner Fall: (a v b) & (~a v c) => b v c

 [klausel(C1),entferne(P,C1,CA),
 klausel(C2),entferne(~P,C2,CB),not fertig(C1,C2,P)] --->
   [assert(klausel(CA v CB)),assert(fertig(C1,C2,P)),write('klausel('),write(C1),
    write(') & klausel('),write(C2),write(') => '),write(klausel(CA v CB)),nl].

[] ---> [write('Kein Widerspruch in der Negation entdeckt!'),nl,stop].

entferne(X,X v Y,Y).      % Entfernt X aus X v Y. Übrig bleibt Y.
entferne(X,Y v X,Y).      %    "        "              "
entferne(X,Y v Z,Y v Z1) :- entferne(X,Z,Z1).     % Entfernt Teilausdruck
entferne(X,Y v Z,Y1 v Z) :- entferne(X,Y,Y1).     %     "          "

in(X,X).                        % X ist in sich selbst enthalten.
in(X,Y) :- entferne(X,Y,_).     % X ist in Y enthalten, wenn es entfernt werden
                                % könnte.
%----------------------------------------------------------------------------
% automatisches Beweisen

beweise(Aussage) :- ausgabe(Gerät),tell(Gerät),write('Zu beweisen: '),
   write(Aussage),nl,nl,nl,write('Negation: '),write(~Aussage),nl,nl,
   übersetze(~Aussage),start.

ausgabe(printer).    % Auf diesem Gerät soll die Ausgabe des Beweises
                     % erfolgen. Es sind alle legalen Datenströme (also
                     % auch Dateien) möglich.

end.
```

Listing 1: Definition einiger Operatoren

kann damit also so etwas wie ein natürliches Modell für parallele Datenverarbeitung angesehen werden, da jedes Modul auf einem separaten Prozessor implementiert werden kann, der auf die gemeinsame Datenbasis wirkt. Die Erfüllung von Bedingungsmustern erinnert natürlich sofort an den PROLOG-Unifizierungsprozeß. In der Tat ist der PROLOG-Interpreter nichts anderes als ein musterorientiertes Programmsystem. Im Beispielprogramm Listing 1 werden die Module in der Form

[bedingungen] → [aktionen]

dargestellt. Hierin bedeutet [bedingungen] eine Liste von Bedingungen in klausaler Form und [aktionen] eine Liste mit Prädikaten, die im Falle eines erfolgreichen Tests ausgeführt werden sollen. Das Prädikat **start** setzt dann das musterorientierte Programm in Aktion, indem es das goal **Bedingungen → Aktionen** im Prädikat **start** zu unifizieren versucht. Hat der PROLOG-Interpreter eine Instanz gefunden, dann wird ein Test ausgeführt, der alle Klauseln in der Bedingungsliste zu unifizieren versucht. Ist das gelungen, dann werden die in der Liste **Aktionen** aufgeführten Prädikate zur Unifizierung gebracht. Wenn alle Prädikate unifiziert worden sind, bleibt nur noch die leere Liste als Aktionsliste zurück. Dann wird durch erneuten Aufruf von **start** nach dem nächsten Muster in der Datenbasis gesucht. Ist kein weiteres Muster vorhanden, hat der PROLOG-Interpreter einen Mißerfolg und beendet das Prädikat **start.**

Das Prädikat op/3

Zur Funktionsweise des Programms in Listing 1 trägt in ganz besonderer Weise die Möglichkeit bei, in PROLOG eigene Operatoren zu vereinbaren. Dies geschieht mit Hilfe des Prädikates **op/3**. Die Syntax des Aufrufs ist **op(Priorität,Assoziativität,Name).** Hierin bedeutet **Priorität** eine Zahl zwischen 1 und 1200. Die Bedeutung dieser Zahl wird klar, wenn man sich an die Eselsbrücke *Punktrechnung geht vor Strichrechnung* erinnert. In PROLOG erkennt man diese Eselsbreücke in Form der Vereinbarung **op(500,yfx,+)** bzw. **op(400,yfx,★)** wieder. Man sieht: Je



Kleiner die Priorität ist, desto eher wird der entsprechende Operator ausgewertet.

Die Assoziativität beschreibt gleichzeitig die Stelligkeit und Position des neu definierten Operators. Die Assoziativität darf nur einen der folgenden Werte haben:

fx, fy, xfx, xfy, yfx, yfy, xf, yf

f steht für Funktor und kennzeichnet seine Stellung innerhalb eines Terms: fx und fy geben an, daß der neue Operator in Präfix-Notation zu gebrauchen ist. xfx, xfy, yfy und yfx kennzeichnen einen in Infix-Notation zu verwendenen Operator. xf und yf schließlich sind die Assoziativität eines Postfix-Operators. Wie man sieht, sind + und ★ also Infix-Operatoren, d. h. PROLOG erwartet das Additionszeichen bzw. Multiplikationszeichen **zwischen** zwei Zahlen oder Termen. Und was bedeutet x und y? Ein **x** gibt an, daß auf der betreffenden Seite alle Terme Operatoren mit niedrigerer Priorität haben müssen. Ein **y** läßt dagegen auch Operatoren mit niedriger oder gleicher Priorität zu. Um das zu verdeutlichen, schauen wir uns einen beliebigen arithmetischen Term an:

3★X+7−Y/4

Dieser arithmetische Term ist äquivalent zu ((3★X)+7)−(Y/4). Ein Blick in TAb. 1 zeigt uns, warum auch PROLOG diese Interpretation mit uns teilt. Der erste Operator, auf den der Interpreter bei der Bearbeitung des Termes stößt, ist ★. Auf der linken Seite ist alles klar, der einzige Operand ist das numerische Atom **3**. Mehrdeutig könnte es auf der rechten Seite werden, wenn nicht die Assoziativität des ★-Operators den Wert yfx hätte. Damit besteht die Forderung, daß rechts vom ★-Zeichen nur Operatoren kleinerer Priorität stehen dürfen. + hat aber die Priorität 500 – und die ist um 100 größer als die Priorität 400, welche ★ besitzt. Folglich wird nur der Term **3★X** bewertet und berechnet. Das Ergebnis steht nun zur Verfügung und

```
op(+,yfx,500).
op(-,yfx,500).
op(*,yfx,400).
op(/,yfx,400).
```

Tabelle 1

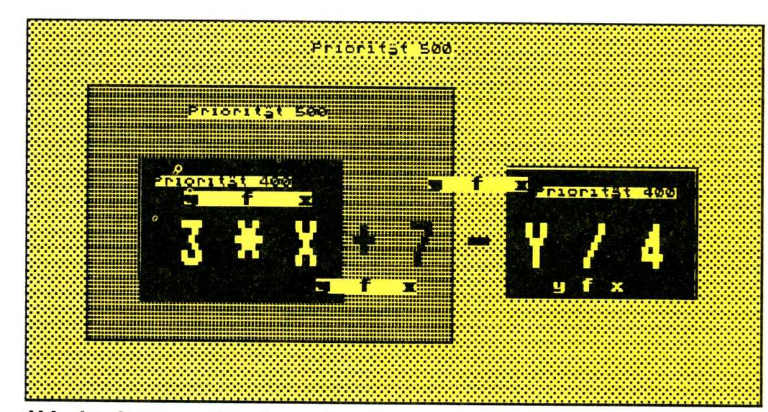


Abb.1: Argumentbindung durch unterschiedliche Prioritäten

der Interpreter nimmt diesen Wert als linke Seite des folgenden +-Operators. Da die rechte Seite nur kleinere **und nicht gleiche** Priorität haben darf, gehört der auf – folgende Term nicht zur rechten Seite. Das Ergebnis ist als (3★X)+7. Dieses Ergebnis wiederum stellt nun die linke Seite des folgenden Subtraktionsoperators dar. Und da / eine niedrigere Priorität besitzt als –, gehört der Term **Y/4** in die rechte Seite des Subtraktionsoperators. Also wird jetzt erst der Term **Y/4** bewertet. Die linke Seite des Divisionsoperators endet bei dem –Zeichen, da dieses eine höhere Priorität besitzt. Also wird jetzt nur der Wert **Y/4** berechnet. Das Ergebnis ist jetzt der rechte Teil des Subtraktionsoperators, der jetzt folgende Operation durchzuführen hat: ((3★X)+7) – (Y/4). Abb. 1 zeigt diese Situation nochmals graphisch.

Im Programm Listing 1 werden nun die folgenden Operatoren definiert:

→ Der Aktionsoperator für musterorientierte Programmierung. Er trennt die Bedingungsliste von der Aktionsliste.

~ Der Negationsoperator. Negiert die logischen Terme.

& Der Konjunktionsoperator. Verknüpfung zweier Terme durch logisches *und.*

v Der Disjunktionsoperator. Verknüpft zwei logische Terme durch *oder.*

→ Der Implikationsoperator a → b heißt: Wenn a gilt, dann gilt auch b.

Die Prioritäten der entsprechenden Operatoren ergeben sich entsprechend den Regeln der Logik, wonach die Negation vor der Konjunktion vor der Disjunktion vor der Implikation zu berücksichtigen ist. Die Negation wird als Präfixoperator definiert (~ steht immer **vor** dem logischen Term, der negiert wird), die übrigen Operatoren werden als Infixoperatoren definiert (d. h. sie stehen **zwischen** zwei Termen). Nach der Definiton dieser Operatoren ist PROLOG in der Lage, logische Terme korrekt zu bearbeiten.

Das Prädikat forma um/2

Diese Prädikate enthalten das Wissen um die Umformung logischer Terme. So bedeutet z. B. **forme_um(~(~X), X).**, daß der Term ~(~X) durch den vereinfachten Term X ersetzt werden kann (Prinzip der doppelten Negation). Ähnliche Umformungen finden sich für die Eliminierung der Implikation, die De Morgan'schen Gesetze und die Distributivgesetze. Falls ein Term noch Unterterme enthält, die umgeformt werden können, sorgen die drei letzten Instanzen dieses Prädikates für die entsprechende Umformung. Die Übersetzung eines Termes in Klauselform wird durch das Prädikat **übersetze/1** vorgenommen. Dieses übersetzt einen Term in der Klauselform, indem es jeden einzelnen Term umformt. Ist keine weitere Umformung nach den Regeln **forme_um** mehr möglich, wird die so gefundene Formel in die Datenbasis aufgenommen.



Die Produktionsregeln für das Resolutionsverfahren

Die Produktionsregeln, nach denen die Terme in konjunktiver Normalform abgearbeitet werden, sind in Form der oben beschriebenen Muster in der Datenbank enthalten. So lautet die Regel, die den Widerspruch in der Datenbank entdeckt:

[klausel(X),klausel(~ X)] → [write('Widerspruch gefunden'),...]

Wenn also in der Datenbank sowohl die Klausel X als auch ihre Negation enthalten ist, dann liegt ein Widerspruch vor und die Aktionsliste druckt die entsprechende Mitteilung aus und stoppt die weitere Ausführung des Programms. In den Produktionsregeln taucht u. a. das Prädikat **fertig/3** auf. Es dient dazu, einen Überlauf in der Datenbank zu verhindern. Die Resolution produziert ja neue Klauseln: (a v b) & (~ a v c) → (b v c). Bei Existenz zweiter Klauseln der Form (a v b) & (~ a v c) wird folglich eine neue Klausel (b v c) produziert und in die Datenbank aufgenommen. Da aber die alten Klauseln (a v b) und (~ a v c) nicht aus der Datenbank entfernt werden, würde beim nächsten Aufruf der Resolution wieder die gleiche Klausel produziert werden. Um das zu verhindern, wird bei der ersten Produktion einer Klausel das Prädikat **fertig((a v b),(~ a v c),a).** in die Datenbank aufgenommen, was eine nochmalige Resolution obiger Klauseln verhindert, die Resolution von Untertermen aber noch zuläßt. Falls schließlich alle logischen Terme in der Datenbasis umgeformt sind und keine weitere Resolution mehr möglich ist, dann trifft PROLOG schließlich auf die leere Bedingungsliste und weiß, daß kein Widerspruch in der Datenbasis gefunden wurde. In diesem Fall ist der Beweis des zu beweisenden Satzes gescheitert! Zur Übung sollten Sie sich den Beweis (a → b)&(b → c) → (a → c) einmal vom Programm vorführen lassen. Anschließend schauen wir uns die Arbeitsweise des Programms einmal an einem berühmten Beispiel an: Der Frage nämlich, ob Marcus Cäsar haßte.

```
:- beweise((mann(marcus) & pompejaner(marcus) & (pompejaner(X) => römer(X)) &
herrscher(cäsar) & (römer(X) => (loyal(X,cäsar) v haßt(X,cäsar))) & (mann(X) v
herrscher(Y) v versucht_mord(X,Y) => ~ loyal(X,Y)) &
versucht_mord(marcus,cäsar)) => haßt(marcus,cäsar)).
```

Listing 2: Die geschichtliche Frage an den Computer

Haßte Marcus Cäsar?

Dieses Beispiel stammt von Elaine Rich und ist eine gute Demonstration für die Instanzierung von Variablen in logischen Termen während des Beweises. Hier die Postulate, von denen im Beweis ausgegangen wird:

mann(marcus). Marcus war ein Mann.

pompejaner(marcus). Marcus war ein Pompejaner.

pompejaner(X) → römer(X). Alle Pompejaner waren Römer.

herrscher(cäsar). Cäsar war ein Herrscher.

römer(X) → loyal(X,cäsar) v haßt (X,cäsar). Alle Römer waren entweder Cäsar gegenüber loyal oder sie haßten ihn.

mann(X) & herrscher(Y) & versucht_mord(X,Y) → ~ loyal(X,Y).
Wenn ein Mensch einen Herrscher zu ermorden versucht, dann ist er ihm nicht loyal gegenüber.

versucht_mord(marcus,cäsar). Markus versuchte Cäsar zu ermorden.

Listing 2 zeigt nun die Eingabe, die für den Beweis nötig ist, daß Marcus Cäsar haßte, und Abb. 2 zeigt die Arbeit des Programmes.

(Dr. Sarnow)

Literaturhinweise:

1.) Bratko, Ivan. PROLOG Programming for artificial Intelligence. Addison Wesley Publishing. 1986.

2.) Hofstadter, Douglas R. Gödel, Escher, Bach: ein endlos geflochtenes Band. Ernst Klett Verlage. 1985.

3.) Rich, Elaine. Artificial Intelligence. McGraw-Hill 1983.


```
Zu beweisen: mann(marcus) & pompejaner(marcus) & (pompejaner(X1) => römer(X1)) & herrscher(cäsar) & (römer(X1) => loyal(X1, cäsar) v haßt(X1, cäsar)) & (mann(X1) v herrscher(X2) v versucht_mord(X1, X
2) => ~ loyal(X1, X2)) & versucht_mord(marcus, cäsar) => haßt(marcus, cäsar)

Negation: ~ (mann(marcus) & pompejaner(marcus) & (pompejaner(X1) => römer(X1)) & herrscher(cäsar) & (römer(X1) => loyal(X1, cäsar) v haßt(X1, cäsar)) & (mann(X1) v herrscher(X2) v versucht_mord(X1,
X2) => ~ loyal(X1, X2)) & versucht_mord(marcus, cäsar) => haßt(marcus, cäsar))

( ~ (mann(marcus) & pompejaner(marcus) & (pompejaner(X1) => römer(X1)) & herrscher(cäsar) & (römer(X1) => loyal(X1, cäsar) v haßt(X1, cäsar)) & (mann(X1) v herrscher(X2) v versucht_mord(X1, X2) => ~
 loyal(X1, X2)) & versucht_mord(marcus, cäsar) => haßt(marcus, cäsar))) => ( ~ ( ~ (mann(marcus) & pompejaner(marcus) & (pompejaner(X1) => römer(X1)) & herrscher(cäsar) & (römer(X1) => loyal(X1, cäsa
r) v haßt(X1, cäsar)) & (mann(X1) v herrscher(X2) v versucht_mord(X1, X2) => ~ loyal(X1, X2)) & versucht_mord(marcus, cäsar)) v haßt(marcus, cäsar)))
( ~ ( ~ (mann(marcus) & pompejaner(marcus) & (pompejaner(X1) => römer(X1)) & herrscher(cäsar) & (römer(X1) => loyal(X1, cäsar) v haßt(X1, cäsar)) & (mann(X1) v herrscher(X2) v versucht_mord(X1, X2) =
> ~ loyal(X1, X2)) & versucht_mord(marcus, cäsar)) v haßt(marcus, cäsar))) => ( ~ ~ (mann(marcus) & pompejaner(marcus) & (pompejaner(X1) => römer(X1)) & herrscher(cäsar) & (römer(X1) => loyal(X1, c
äsar) v haßt(X1, cäsar)) & (mann(X1) v herrscher(X2) v versucht_mord(X1, X2) => ~ loyal(X1, X2)) & versucht_mord(marcus, cäsar)) & ~ haßt(marcus, cäsar))
( ~ ~ (mann(marcus) & pompejaner(marcus) & (pompejaner(X1) => römer(X1)) & herrscher(cäsar) & (römer(X1) => loyal(X1, cäsar) v haßt(X1, cäsar)) & (mann(X1) v herrscher(X2) v versucht_mord(X1, X2) =>
 ~ loyal(X1, X2)) & versucht_mord(marcus, cäsar))) => (mann(marcus) & pompejaner(marcus) & (pompejaner(X1) => römer(X1)) & herrscher(cäsar) & (römer(X1) => loyal(X1, cäsar) v haßt(X1, cäsar)) & (man
n(X1) v herrscher(X2) v versucht_mord(X1, X2) => ~ loyal(X1, X2)) & versucht_mord(marcus, cäsar))
klausel(mann(marcus))
klausel(pompejaner(marcus))
(pompejaner(X1) => römer(X1)) => ( ~ pompejaner(X1) v römer(X1))
klausel( ~ pompejaner(X1) v römer(X1))
klausel(herrscher(cäsar))
(römer(X1) => loyal(X1, cäsar) v haßt(X1, cäsar)) => ( ~ römer(X1) v loyal(X1, cäsar) v haßt(X1, cäsar))
klausel( ~ römer(X1) v loyal(X1, cäsar) v haßt(X1, cäsar))
(mann(X1) v herrscher(X2) v versucht_mord(X1, X2) => ~ loyal(X1, X2)) => ( ~ (mann(X1) v herrscher(X2) v versucht_mord(X1, X2)) v ~ loyal(X1, X2))
( ~ (mann(X1) v herrscher(X2) v versucht_mord(X1, X2)) v ~ loyal(X1, X2)) => ( ~ mann(X1) & ~ (herrscher(X2) v versucht_mord(X1, X2)) v ~ loyal(X1, X2))
( ~ mann(X1) & ~ (herrscher(X2) v versucht_mord(X1, X2)) v ~ loyal(X1, X2)) => (( ~ mann(X1) v ~ loyal(X1, X2)) & ( ~ (herrscher(X2) v versucht_mord(X1, X2)) v ~ loyal(X1, X2)))
klausel( ~ mann(X1) v ~ loyal(X1, X2))
( ~ (herrscher(X1) v versucht_mord(X2, X1)) v ~ loyal(X2, X1)) => ( ~ herrscher(X1) & ~ versucht_mord(X2, X1) v ~ loyal(X2, X1))
( ~ herrscher(X1) & ~ versucht_mord(X2, X1) v ~ loyal(X2, X1)) => (( ~ herrscher(X1) v ~ loyal(X2, X1)) & ( ~ versucht_mord(X2, X1) v ~ loyal(X2, X1)))
klausel( ~ herrscher(X1) v ~ loyal(X2, X1))
klausel( ~ versucht_mord(X1, X2) v ~ loyal(X1, X2))
klausel(versucht_mord(marcus, cäsar))
klausel( ~ haßt(marcus, cäsar))
klausel( ~ haßt(marcus, cäsar)) & klausel( ~ römer(marcus) v loyal(marcus, cäsar) v haßt(marcus, cäsar)) => klausel( ~ römer(marcus) v loyal(marcus, cäsar))
klausel( ~ römer(marcus) v loyal(marcus, cäsar)) & klausel( ~ versucht_mord(marcus, cäsar) v ~ loyal(marcus, cäsar)) => klausel( ~ römer(marcus) v ~ versucht_mord(marcus, cäsar))
klausel(versucht_mord(marcus, cäsar)) & klausel( ~ römer(marcus) v ~ versucht_mord(marcus, cäsar)) => klausel( ~ römer(marcus))
klausel( ~ römer(marcus)) & klausel( ~ pompejaner(marcus) v römer(marcus)) => klausel( ~ pompejaner(marcus))
Widerspruch in der Negation gefunden!
Aussage damit wahr.
```

Abbildung 2: Haßte Markus Cäsar? Ein neuzeitlicher Lösungsweg!

CAD MIT CAMPUS?

Eine CAD-Sensation für 798 DM oder nur ein komfortables Malprogramm? Wir testeten das CAD-Paket „Campus" von Digital Workshop für den Atari ST.

Die junge Bochumer Softwareschmiede Digital Workshop bietet seit Anfang des Jahres ein preisgünstiges CAD-Programm an, das durch seinen hohen Anspruch (Handbuch: „...*ausschließlich* in Richtung Computer Aided Konstruktion mit den Schwerpunkten Entwerfen und Konstruieren") und seinen Tiefstapelpreis – die für den Test vorliegende Version 1.1 ist für nur 798 DM im Handel – neugierig macht.

CAD (= **C**omputer **A**ided **D**esign) hat sich mittlerweile zu einem Sammelbegriff entwickelt, der *alle* Aktionen beschreibt, bei denen mittel- oder unmittelbar im Rahmen von Entwicklungs- und Konstruktionsaufgaben der Rechnereinsatz eingebunden ist. CAD darf sich also nicht nur auf das bloße Erstellen zweidimensionaler Zeichnungen beschränken. Ein CAD-Programm – hier also Campus – muß erheblich mehr leisten können als etwa die bekannten Malprogramme (DEGAS, STAD, Monostar etc.).

Billig oder preiswert?

PC-CAD-Software – unter MS-DOS auf IBM-kompatiblen PCs lauffähig – kostet im Schnitt um die 12 000 DM pro Paket. Bekannteste Vertreter sind Autocad von Autodesk und, als deutsches Produkt, PC-Draft von rhv. Die Kosten für Campus sind demgegenüber geradezu unverschämt gering.

Fragen über Fragen: Ist Campus ein „richtiges" CAD-Programm? Ist es im Vergleich mit anderen preiswert – oder nur billig? Und schließlich: Ist der Vergleich von Campus mit anderen CAD-Systemen angesichts des enormen Preisunterschieds überhaupt fair?

Lieferung und Leistung

Campus (übrigens in C geschrieben) wird auf zwei Disketten im praktischen DIN A5-formatigen Handbuch geliefert. Erstes Lob: Auf den für die Vorläuferversionen noch notwendigen Hardwareschutz (Stecker für den ROM-Port) wurde mittlerweile verzichtet.

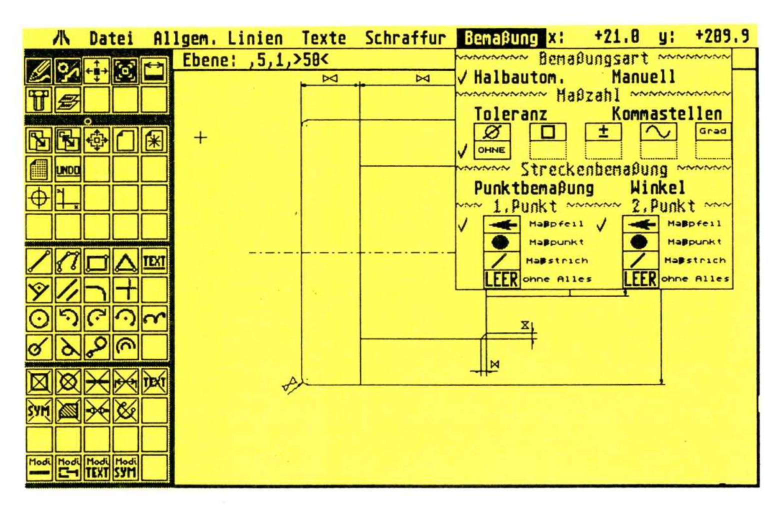


Abbildung 1

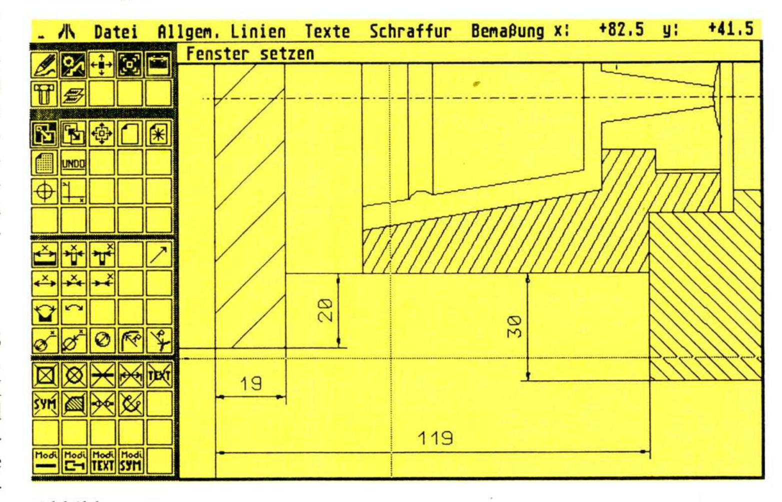
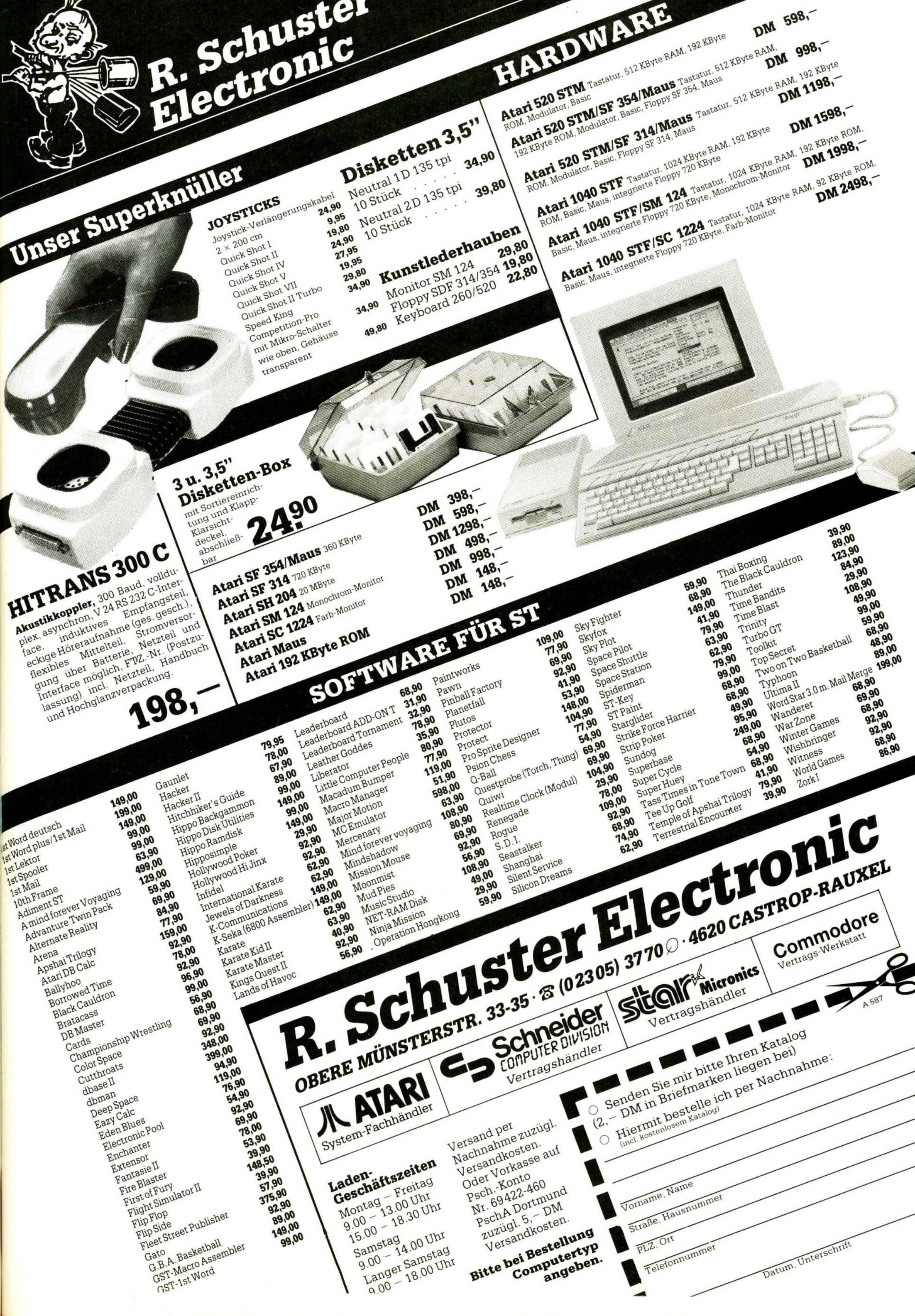


Abbildung 2

Für den Betrieb von Campus ist folgende Hardwarekonfiguration Voraussetzung: ST mit mindestens 512 KB RAM, TOS im ROM, ein Floppy-Laufwerk, Monochrom-Monitor, Maus, Drucker und/oder Plotter als Ausgabegerät.

Die Aufrüstung des Arbeitsspeichers auf 1 MB ist natürlich ebenso empfehlenswert wie die Verwendung einer Festplatte. Wichtiger Hinweis im Handbuch: Digital Workshop bietet einen Up-date-Service an. Jeder Campus-Käufer wird – falls gewünscht – mit aktuellen Informationen versorgt und kann die jeweils neueste Programmversion unter Anrechnung des Kaufpreises (mindestens sind 30 DM zu zahlen) beziehen. Zur Zeit wird für die registrierten Campus-Kunden über eine Hot-line Telefonsupport gewährt. Um mit Campus arbeiten zu können, muß man selbstverständlich ausreichendes Know-how



R. Schuster Electronic
HARDWARE
Atari 520 STM Tastatur, 512 KByte RAM, 192 KByte ROM, Modulator, Basic
DM 598,–
Atari 520 STM/SF 354/Maus Tastatur, 512 KByte RAM, 192 KByte ROM, Modulator, Basic, Floppy SF 354, Maus
DM 998,–
Atari 520 STM/SF 314/Maus Tastatur, 512 KByte RAM, 192 KByte ROM, Modulator, Basic, Floppy SF 314, Maus
DM 1198,–
Atari 1040 STF Tastatur, 1024 KByte RAM, 192 KByte ROM, Basic, Maus, integrierte Floppy 720 KByte
DM 1598,–
Atari 1040 STF/SM 124 Tastatur, 1024 KByte RAM, 192 KByte ROM, Basic, Maus, integrierte Floppy 720 KByte, Monochrom-Monitor
DM 1998,–
Atari 1040 STF/SC 1224 Tastatur, 1024 KByte RAM, 92 KByte ROM, Basic, Maus, integrierte Floppy 720 KByte, Farb-Monitor
DM 2498,–
Unser Superknüller
JOYSTICKS
Joystick-Verlängerungskabel 2 × 200 cm
Quick Shot I
Quick Shot II
Quick Shot IV
Quick Shot V
Quick Shot VII
Quick Shot II Turbo
Speed King
Competition-Pro mit Mikro-Schalter
wie oben, Gehäuse transparent
24,90
9,95
19,80
24,90
27,95
19,95
29,80
34,90
34,90
49,80
Disketten 3,5"
Neutral 1 D 135 tpi 10 Stück 34,90
Neutral 2 D 135 tpi 10 Stück 39,80
Kunstlederhauben
Monitor SM 124 29,80
Floppy SDF 314/354 19,80
Keyboard 260/520 22,80
3 u. 3,5" Disketten-Box
mit Sortiereinrichtung und Klapp-Klarsichtdeckel, abschließbar
24,90
HITRANS 300 C
Akustikkoppler, 300 Baud, vollduplex, asynchron, V 24 RS 232 C-Interface, induktives Empfangsteil, eckige Höreraufnahme (ges. gesch.), flexibles Mittelteil, Stromversorgung über Batterie, Netzteil und Interface möglich, FTZ.-Nr. (Postzulassung) incl. Netzteil, Handbuch und Hochglanzverpackung.
198,—
Atari SF 354/Maus 360 KByte
Atari SF 314 720 KByte
Atari SH 204 20 MByte
Atari SM 124 Monochrom-Monitor
Atari SC 1224 Farb-Monitor
Atari Maus
Atari 192 KByte ROM
DM 398,–
DM 598,–
DM 1298,–
DM 498,–
DM 998,–
DM 148,–
DM 148,–
SOFTWARE FÜR ST
1st Word deutsch
1st Word plus/1st Mail
1st Lektor
1st Spooler
1st Mail
10th Frame
Adiment ST
A mind forever Voyaging
Adventure Twin Pack
Alternate Reality
Arena
Apshai Trilogy
Atari DB Calc
Ballyhoo
Borrowed Time
Black Cauldron
Bratacass
DB Master
Cards
Championship Wrestling
Color Space
Cutthroats
dbase II
dbman
Deep Space
Eazy Calc
Eden Blues
Electronic Pool
Enchanter
Extensor
Fantasie II
Fire Blaster
First of Fury
Flight Simulator II
Flip Flop
Flip Side
Fleet Street Publisher
Gato
G.B.A. Basketball
GST-Macro Assembler
GST-1st Word
149,00
199,00
149,00
99,00
99,00
63,90
499,00
129,00
59,90
69,90
84,90
77,90
159,00
92,90
78,00
92,90
96,90
99,00
56,90
68,90
69,90
92,90
348,00
399,00
94,90
119,00
76,90
54,90
92,90
69,90
78,00
53,90
39,90
148,50
39,90
57,90
375,90
92,90
89,00
149,00
99,00
Gaunlet
Hacker
Hacker II
Hitchhiker's Guide
Hippo Backgammon
Hippo Disk Utilities
Hippo Ramdisk
Hipposimple
Hollywood Poker
Hollywood Hi Jinx
Infidel
International Karate
Jewels of Darkness
K-Communications
K-Seka (6800 Assembler)
Karate
Karate Kid II
Karate Master
Kings Quest II
Lands of Havoc
79,95
78,00
67,90
89,00
99,00
149,00
99,00
149,00
29,90
92,90
92,90.
62,90
62,90
149,00
149,00
62,90
63,90
40,90
92,90
56,90
Leaderboard
Leaderboard ADD-ON T
Leaderboard Tornament
Leather Goddes
Liberator
Little Computer People
Macadum Bumper
Macro Manager
Major Motion
MC Emulator
Mercenary
Mind forever voyaging
Mindshadow
Mission Mouse
Moonmist
Mud Pies
Music Studio
NET-RAM Disk
Ninja Mission
Operation Hongkong
68,90
31,90
32,90
78,90
35,90
80,90
77,90
119,00
51,90
598,00
63,90
108,90
80,90
69,90
92,90
56,90
108,90
49,00
29,90
59,90
Paintworks
Pawn
Pinball Factory
Planetfall
Plutos
Protector
Protect
Pro Sprite Designer
Psion Chess
Q-Ball
Questprobe (Torch, Thing)
Quiwi
Realtime Clock (Modul)
Renegade
Rogue
S.D.I.
Seastalker
Shanghai
Silent Service
Silicon Dreams
109,00
77,90
69,90
92,90
41,90
53,90
148,00
104,90
77,90
54,90
69,90
69,90
104,90
29,90
78,00
109,00
92,90
68,90
74,90
62,90
Sky Fighter
Skyfox
Sky Plot
Space Pilot
Space Shuttle
Space Station
Spiderman
ST-Key
ST Paint
Starglider
Strike Force Harrier
Strip Poker
Sundog
Superbase
Super Cycle
Super Huey
Tass Times in Tone Town
Tee Up Golf
Temple of Apshai Trilogy
Terrestrial Encounter
59,90
68,90
149,00
41,90
79,90
63,90
62,90
79,90
99,00
68,90
68,90
49,90
95,90
249,00
68,90
54,90
68,90
41,90
79,90
39,90
Thai Boxing
The Black Cauldron
Thunder
Time Bandits
Time Blast
Trinity
Turbo GT
Toolkit
Top Secret
Two on Two Basketball
Typhoon
Ultima II
Word Star 3.0 m. Mail Merge
Wanderer
War Zone
Winter Games
Wishbringer
Witness
World Games
Zork I
39,90
89,00
123,90
84,90
29,90
108,90
49,90
99,00
59,90
68,90
48,90
89,00
199,00
68,90
69,90
68,90
92,90
92,90
68,90
86,90
R. Schuster Electronic
OBERE MÜNSTERSTR. 33-35 · ☎ (02305) 3770 · 4620 CASTROP-RAUXEL
ATARI System-Fachhändler
Schneider COMPUTER DIVISION Vertragshändler
star Micronics Vertragshändler
Commodore Vertrags-Werkstatt
Laden-Geschäftszeiten
Montag – Freitag
9.00 – 13.00 Uhr
15.00 – 18.30 Uhr
Samstag
9.00 – 14.00 Uhr
Langer Samstag
9.00 – 18.00 Uhr
Versand per Nachnahme zuzügl. Versandkosten.
Oder Vorkasse auf Psch.-Konto Nr. 69422-460 PschA Dortmund zuzügl. 5,– DM Versandkosten.
Bitte bei Bestellung Computertyp angeben.
○ Senden Sie mir bitte Ihren Katalog (2,– DM in Briefmarken liegen bei)
○ Hiermit bestelle ich per Nachnahme: (incl. kostenlosem Katalog)
Vorname, Name
Straße, Hausnummer
PLZ, Ort
Telefonnummer
Datum, Unterschrift
A 587

über den Umgang mit dem Atari ST besitzen.

Ans Eingemachte

Für den Test stand ein auf 1 MB aufgerüsteter Atari ST mit doppeltem Laufwerk zur Verfügung. Campus belegt ca. 300 KB im Arbeitsspeicher und ist innerhalb von 40 sec. geladen. Nach dem Start des Programms wird dem Benutzer eine Menüleiste präsentiert, die die Begriffe „Ende", „Neu" (neue Zeichnung erstellen) und „Laden" (bereits erstellte Zeichnung laden) enthält.

Die Auswahl „Neu" verlangt u. a. eine Festlegung der gewünschten Größe der Zeichenfläche (DIN A4 – DIN A0 stehen zur Verfügung, jeweils hochkant oder quer) sowie des Maßstabes, in dem die Zeichnung erstellt werden soll. Leider sind diese einmal gewählten Parameter für die gesamte Bearbeitung bindend; sie können im Verlauf der Konstruktion nicht mehr geändert werden.

Campus arbeitet mit der GEM-Oberfläche. Abb. 1 und und 2 zeigen u. a. die Aufteilung des Bildschirms. Alle Einstellungen und Befehle werden mit der Maus aktiviert. Die Tastatur dient lediglich zur Eingabe von Koordinaten.

In der obersten Bildschirmzeile sind die Drop-down-Menüs untergebracht (Parametereinstellung, Dateiverwaltung etc.); das linke Bildschirmviertel beherbergt den gesamten Befehlsvorrat des Systems in Symboldarstellung. Die obersten zwei Reihen dieses Bereichs stellen das Hauptmenü dar. Zur Auswahl stehen derzeit zur Verfügung (7 von 10 Feldern sind belegt):

- Zeichenfunktionen
- Löschen und Ändern
- Bewegungsfunktionen (Transformieren von Zeichnungsinhalten wie Kopieren, Verkleinern, Vergrößern, Spiegeln und Drehen)
- Blattfunktionen (Bildschirmbezogene Darstellung der Zeichnung, Zoomen etc., Raster)
- Bemaßung
- Symbole (Generieren und Einbinden von mehrmals wiederverwendbaren Zeichnungsteilen wie Schrauben, Bearbeitungszeichen etc.)
- Ebenen

Unter den zwei Hauptmenüzeilen sind drei Blöcke à 20 Felder zur Aufnahme der Symbole für die jeweils ausgewählten Befehlsgruppen vorgesehen. Im linken Bildschirmviertel lassen sich also außer den beiden Hauptmenüzeilen drei vollständige Befehlsgruppen gleichzeitig darstellen (vgl. Abb. 1: Zeichenfunktionen, Löschen und Ändern, Blattfunktionen). Diese Benutzerführung ist zwar komfortabel, bedingt aber durch die Informationsfülle auf dem Bildschirm einen sehr kleinen Arbeitsbereich.

Nun geht's ans Eingemachte, denn nun werden die einzelnen derzeit verfügbaren Befehlsgruppen unter die Lupe genommen.

Allgemeine Zeichenfunktionen

Campus erlaubt im wesentlichen das Zeichnen von einzelnen Strecken, Polygonen, Kreisen und Kreisbögen – natürlich in unterschiedlichen Strichstärken (sprich unterschiedlichen Stiften beim Plotten der Zeichnung) und in verschiedenen Linienarten. Wie bei jedem CAD-System müssen für die Start- bzw. Endpunkte von Linien absolute, relative oder polare Koordinaten eingegeben werden. Dies geschieht normalerweise entweder mit der Maus *oder* via Tastatur. Bei Campus dagegen ist auch eine gemischte Eingabe möglich: Beispielsweise wird das Zeichnen einer horizontalen Strecke durch die Eingabe y=0 für den Endpunkt der Strecke per Tastatur *plus* der Angabe einer Position mittels Maus bewirkt. Leider ist diese Möglichkeit der Eingabe auf Normalkoordinaten beschränkt. Wie leicht könnte man Hilfslinien mit beliebigem Winkel zeichnen, wenn die gemischte Eingabe auch bei Polarkoordinaten möglich wäre! Wichtige Konstruktionselemente wie einzelne Punkte, vor allem aber Ellipsen oder Splines (Stützpunktkurven) sind in der derzeitigen Version noch nicht verfügbar.

Hit-Codes

Um mit dem Fadenkreuz einzelne Punkt eder Konstruktion exakt ansprechen zu können, besteht bei Campus die Möglichkeit, einen Fangradius einzuschalten: Bei der Ermittlung eines Punktes wird dann die nähere Umgebung des Fadenkreuzes nach bereits vorhandenen Punkten abgesucht; der nächstliegende Punkt wird automatisch als Konstruktionspunkt definiert.

Es gibt keine Möglichkeit, gezielt nur Mittelpunkte von Strecken oder Kreisbögen, Schnitt- oder Endpunkte von Strecken, Punkte auf Linien in definiertem Abstand zu Linienendpunkten etc., also sogenannte Hit-Codes, anzusprechen. Hit-Codes und deren sinnvolle Anwendung sind aber gerade beim Konstruieren die „Seele vom Geschäft".

Darüberhinaus funktionierte während des Tests der Befehl Fangradius nicht einwandfrei. Ein echter Wermutstropfen!

Editier- und Löschfunktionen

Die Löschfunktionen sind ausreichend, allerdings nur bei korrekter Funktion des Objektfangs auch sinnvoll anzuwenden. Ein Abschnitt einer Strecke – z. B. zwischen zwei Schnittpunkten – läßt sich mit dem Befehl „Strecke zwischen zwei Punkten löschen" eben nur dann exakt löschen, wenn die beiden Punkte auch exakt identifiziert werden. Zum Editieren einer Strecke steht ein Trimm-Befehl zur Verfügung. Er erlaubt das Verlängern oder Verkürzen einer Linie bis zu einer anderen Linie, bzw. das Verlängern oder Verkürzen zweier Linien auf den gemeinsamen Schnittpunkt.

Achtung: Trimmen bezieht sich leider nur auf Strecken, nicht auch auf Kreisbögen! Das Auftrennen von Linien (als logische Unterfunktion von Trimmen) ist nicht möglich.

Bildschirmfunktionen

Die zur Verfügung stehenden Funktionen sind insgesamt ausreichend. Der Benutzer kann sich Teilbereiche seiner Zeichnung vergrößern lassen, Fenster verschieben, das Bild neu aufbauen oder auf volles Format schalten. Wünschenswert wäre ein Befehl, bei anderen CAD-Systemen als „Zoom alles" oder „automatisches Zoomen" bekannt, der den Zeichnungsinhalt in maximal darstellbarer Größe abbildet. Ein Pluspunkt für Campus in dieser Befehlsgruppe ist allerdings die Möglichkeit, den Koordinatenursprung frei auf der Zeichenfläche setzen zu können.

Transformationen

Da alle konstruierten Konturen im

Rechner durch die Koordinaten ihrer Eckpunkte festgelegt sind, lassen sich diese Konturen über Koordinatentransformationen kinderleicht spiegeln, drehen, verschieben und kopieren. Campus stellt dafür etliche, im Handling einfache Befehle zur Verfügung.

Anlaß zur Kritik besteht bei folgenden Punkten:

Die Auswahl der zu transformierenden Zeichnungsinhalte erfolgt immer mittels eines Fensters, einzelne Zeichnungselemente können nicht separat angesprochen werden.

Beim Kopieren eines Bereiches kann jeweils nur eine Kopie angefertigt werden.

Schraffur

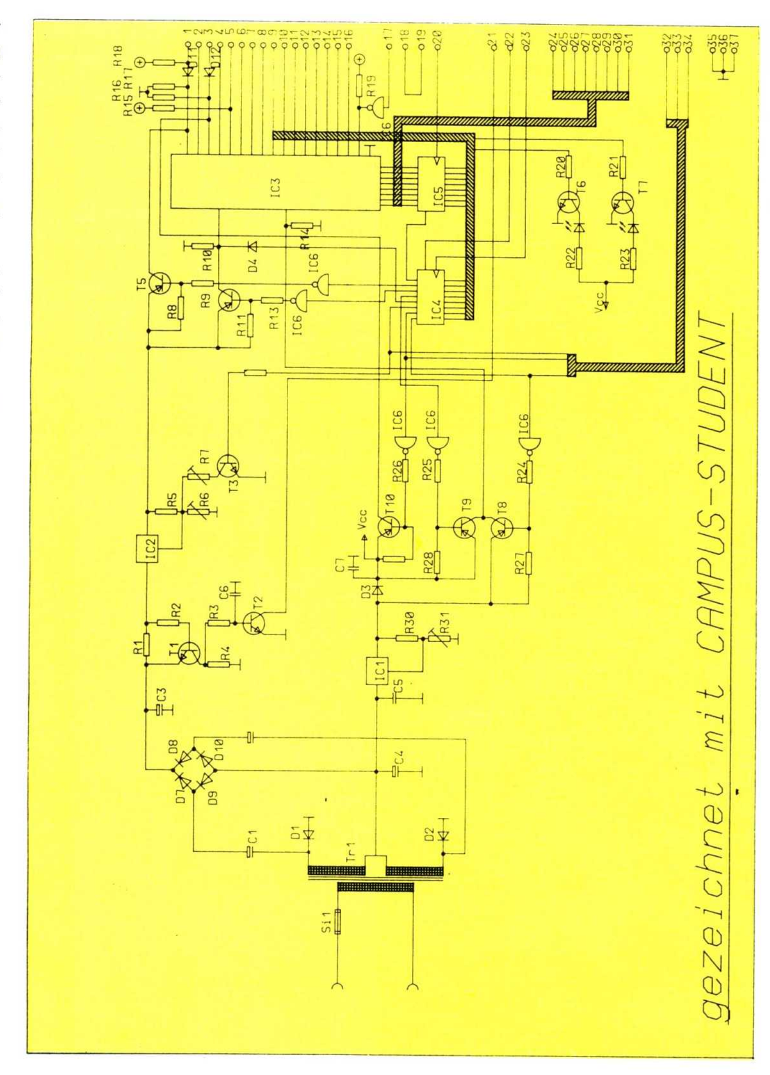
Das System stellt verschiedene Schraffurmuster zur Auswahl. Da nur hundertprozentig geschlossene Konturen schraffiert werden können, ist es vor dem eigentlichen Schraffieren meist erforderlich, die Kontur zu überarbeiten, d. h. sie auf Geschlossenheit zu überprüfen. Hierfür stellt das Programm automatisch eine Schraffurebene zur Verfügung, in der dann nach Herzenslust gelöscht, getrimmt und eventuell zusätzlich gezeichnet werden kann. Diese „Überarbeitungen" werden *nicht* Bestandteil der Zeichnung, sondern dienen einzig der Schraffur. Eine pfiffige Lösung, die allerdings ein einwandfreies Funktionieren des Fangradius voraussetzt.

Text

Hier kann Campus mehrere Pluspunkte verbuchen. Neben verschiedenen Möglichkeiten, Texte in der Zeichnung zu plazieren (einzeilig, mehrzeilig, unter bereits vorhandene Texte, linksbündig, rechtsbündig und zentriert), besitzt das System einen Zeileneditor: Bereits plazierte Texte können damit zeilenweise geändert werden.

Bemaßung

Entsprechend der im Drop-down-Menü gewählten Einheit (vgl. Abb. 1) ist eine halbautomatische Bemaßung möglich. Bemaßt wird eine Strecke bzw. der Abstand zwischen zwei Punkten, d. h. eine Kettenbemaßung muß vom Benutzer aus einzelnen 2-Punkte-Maßlinien zusammengesetzt werden. Die Verwendung des Befehls „Raster" ist hier sehr hilfreich. Natürlich können auch Winkel und Kreisbögen halbautomatisch vermaßt werden. Ab einer bestimmten Abbildungsgröße auf dem Schirm werden die Maßzahlen als zwei Spitze an Spitze liegende Dreiecke dargestellt, um den Benutzer auf eine notwendige Ausschnittsvergrößerung hinzuweisen. Ein wirklich augenschonendes Feature!



Symbol oder auch Makrotechnik

Häufig verwendete Zeichnungsteile wie Schrauben, Schweißzeichen oder Elektrosymbole lassen sich in „Bibliotheken" ablegen und bei Bedarf quasi auf Knopfdruck in die aktuelle Zeichnung einbinden. Diese Befehlsgruppe hinterläßt einen insgeamt guten Eindruck, allerdings mit einer Ausnahme: Um ein Symbol – bei anderen Software-Herstellern auch „Makro" genannt – zu definieren, muß es mittels eines Fensters auf der Zeichnung ausgewählt werden. Differenziertere Auswahlmöglichkeiten wären wünschenswert (vgl. Transformationen).

Strukturierter Zeichnungsaufbau

Campus erlaubt, den Zeichnungsinhalt auf bis zu 9999 Ebenen zu verteilen. Es ist zum Beispiel sinnvoll, die Konturen eines Werkstücks auf einer anderen Ebene abzulegen als die Bemaßung. Durch Ausschalten der Be-

maßungsebene etc. hat man dann die Möglichkeit, die gleiche Zeichnung beispielsweise als Brennteilzeichnung zu benutzen. Eine CAD-Zeichnung besteht also sinnvollerweise immer aus mehreren übereinandergelegten „Klarsichtfolien“, von denen je nach Bedarf einzelne an- bzw. ausgeschaltet oder sichtbar/unsichtbar geschaltet werden können. Die Verwaltung von Ebenen ist derzeit die einzige Möglichkeit in Campus, eine Zeichnung zu strukturieren.

Vermißt wird die Möglichkeit, Zeichnungselemente zu sogenannten Gruppen oder „Clumps“ zusammenfassen zu können.

Thema Geschwindigkeit

Unter Berücksichtigung der Tatsache, daß der Test-Atari ST nicht mit einem arithmetischen Coprozessor (für Fließkommaoperationen) ausgestattet war und auch nicht über sonstiges Hardware-Tuning verfügte, erzielte er respektable Zeiten für den Bildaufbau von komplexen Zeichnungen. Respektabel heißt in diesem Zusammenhang ca. 3 Minuten für eine Zeichnung mit 180 KByte Speicherbedarf. CAD auf PCs stößt hier eben hardwarebedingt auf Grenzen.

Unter dem Strich

Campus ist mit Sicherheit erheblich mehr als nur ein komfortables „Malprogramm“; es steht gewissermaßen fest auf der CAD-Schiene, hat aber noch beträchtliche Entwicklungsstrecken zurückzulegen. Verbesserungen der angesprochenen Mängel sind zumindest wünschenswert, teilweise notwendig.

Noch fehlt eine integrierte Programmiersprache, um z. B. Parameter über kurze Kommandos (oder Anklicken von Symbolen? Es sind ja noch viele Felder unbelegt!) einstellen zu können oder um eigene Variantenprogramme zu erstellen. Um CAD-Programme beispielsweise in Fertigungsprozesse einbinden zu können, sind Programmschnittstellen zur Übergabe von Geometriedaten unerläßlich. Treiber für Digitizer, um vorhandene Papierzeichnungen zu digitalisieren, sind für professionelle Systeme ebenso erforderlich wie die Möglichkeit, größere Graphikschirme betreiben zu können.

Digital Workshop hat sich völlig der CAD-Hard- und Software-Entwicklung verschrieben und es bleibt abzuwarten, mit welchem Tempo der Campus-Zug weiterfährt. Die neue Version 1.2 ist jedenfalls schon angekündigt (Auslieferungstermin ab Mitte Mai).

Für welchen Kundenkreis ist Campus zu empfehlen?

Für den professionellen Einsatz (Konstruktion etc.) hat das System derzeit noch zu viele Ecken und Kanten. Selbst der konkurrenzlos günstige Preis für Hard- und Software wird für einen klein- oder mittelständischen Betrieb nicht kostenentscheidend sein. Die Investitionen für einen vergleichbaren CAD-Arbeitsplatz (Industriestandard) belaufen sich auf etwa 20 000 DM (ohne Plotter). Bei einem Abschreibungszeitraum von fünf Jahren bedeutet das – großzügig über den Daumen gepeilt – Kosten in Höhe von ca. 350 DM/Monat für Arbeitsmittel. Demgegenüber stehen Lohn- bzw. Gehaltskosten von ca. 8000 DM pro Monat (inkl. Arbeitgeberanteil). Aus diesen Zahlen läßt sich leicht ersehen, daß hier keine potentiellen Kunden für Campus zu suchen sind.

Wo sonst!

CAD ist eine zukunftsorientierte, expansive Technologie. CAD-Ausbildungszentren mit Kursgebühren für Einsteiger in Höhe von mehreren tausend Mark sind eventuell ein potentielles Einsatzfeld für Campus. Als Einsteigersystem für Schulungszwecke ist Campus vorbehaltlos zu empfehlen: Schließlich kosten Programm *und* Atari ST *zusammen* weniger als die meisten CAD-Grundkurse. Außerdem hat jeder Campus-Kunde die Aussicht auf eine dynamische Programmweiterentwicklung, über die natürlich in dieser Zeitschrift zu gegebener Zeit zu berichten ist.

Fehlerfreie, perfekte Programme sind leider auch im CAD-Bereich nicht erhältlich. Nicht für 15 000, nicht für 5 000 und schon gar nicht für 800 DM.

(Ulrich Oehler)

Bezugsadresse:

Digital Workshop
Kornharpener Str. 122 a
4630 Bochum 1
Tel. 02 34 / 31 13 04

Profitieren Sie von unserem Know-How!

OMIKRON-BASIC



ARITHMETIK
- Rechengenauigkeit bis 19 Stellen bei allen Funktionen
- Rechenbereich bis 5.11 E±4931

GEM
- Komplette GEM-Library
- sämtliche AES- & VDI-Funktionen direkt mit Namen verfügbar
- eigene BITBLIT-Routine
- NEU: jetzt mit Resource Construction Set

EXTRAS
- Masken-INPUT
- SORT-Befehl sortiert beliebige Felder
- Matrizenbefehle
- Compiler ab Frühjahr verfügbar

STRUKTUR
- Prozeduren und mehrzeilige Funktionen
- mit Übergabe- und Rückgabe-Parametern und lokalen Variablen
- REPEAT...UNTIL, WHILE...WEND, mehrzeiliges IF...THEN...ELSE...ENDIF
- Labels bei GOTO, GOSUB, ON...GOTO etc.

KOMPATIBILITÄT
- 99 % MBASIC-kompatibel
- Editor findet Inkompatibilitäten
- dadurch einfachste Anpassung

Editor
- Mit oder ohne Zeilennummern (umschaltbar)
- Drei Schriftgrößen bis 57x128 °
- Frei definierbare Funktionstasten

GESCHWINDIGKEIT
- FIT-Code (FIT = Fast Interpreting Technique)
- Wir kennen keine schnelleren 68000-Fließkommaroutinen
- Volle Integer-Arithmetik
- eigene Disk-Routinen für beschleunigten Dateizugriff

LIEFERUMFANG
- Modul mit OMIKRON-BASIC (wird seitlich eingesteckt)
- Demodiskette mit Runtime-Interpreter, Hilfsprogrammen und vielen Beispielprogrammen
- 180-seitiges, spiralgebundenes deutsches Handbuch
- Alles zus. nur DM 229,– (unverbindl. Preisempfehlung) Handbuch vorab DM 30,–

„Atemberaubende Geschwindigkeit" (DATA WELT 12/86, S. 69)
„Für ernsthafte Programmierer, die effektive und vor allem schnelle Programme erstellen wollen, ..., ist dieser neue BASIC-Interpreter genau das richtige" (DATA WELT 12/86, S. 70)
„OMIKRON-BASIC kann wesentlich mehr als der BASIC-Standard" (ST Computer 12/86, S. 79)

OMIKRON Software · Erlachstr. 15 · 7534 Birkenfeld · Tel. (0 70 82) 53 86

Österreich: Ueberreuter Media, Alser Str. 24, A-1091 Wien
Schweiz: Microtron, Postfach 40, CH-4542 Pieterlen

Textsysteme im Vergleich:
● 1st Word Plus,
● BECKERtext ST
● Signum

Atari für Textverarbeitung

Erfahrungsgemäß werden Personal-Computer am häufigsten zur Textverarbeitung eingesetzt. Gerade hierfür empfiehlt sich der Atari ST aufgrund seines hochauflösenden monochromen Bildschirms und der flimmerfreien Darstellung mit schwarzer Schrift auf weißem Grund. Folgerichtig werden für Textverarbeitung auch die meisten Programmpakete angeboten. Dies nehmen wir zum Anlaß, einmal drei der markantesten Vertreter dieses Genres miteinander zu vergleichen. Dabei haben wir uns auf Textverarbeitung im engeren Sinne beschränkt: Systeme für Desktop Publishing, zum Beispiel Publishing Partner (Text in ST 4/87), oder Satzsysteme wie TeX (Software-Test ist in Vorbereitung) sind eigentlich schon Spezialanwendungen und eher mit Programmiersprachen als noch mit gewöhnlicher Textverarbeitung vergleichbar.

Die Kandidaten sind also: 1st Word Plus in der neuesten, deutschen Version, BECKERtext ST, der mit heißer Nadel genähte Nachfolger von Textomat, und Signum, das für die Eingabe von Formeln bestens eingeführte Pracht exemplar.

1st Word Plus

1st Word Plus wurde bereits in der ST 4/87 getestet, sodaß wir uns hier auf das Nötigste beschränken können. Die wichtigsten Leistungsmerkmale sind:

- Volle Einbindung in GEM mit Darstellung von Schriftattributen auf dem Bildschirm
- Formatierbefehle zur Textgestaltung (Blocksatz, Zentrieren usw.)
- Blockoperationen
- Suchen/Ersetzen



Abbildung 1: Signet von 1st Word Plus



Abbildung 2: Signet von BECKERtext ST

- Individuelle Druckeranpassung ist möglich
- Bearbeitung von bis zu vier Texten gleichzeitig
- Wörterbuch
- Halbautomatische Trennung
- Druckausgabe wird automatisch gespoolt
- Fußnotenverwaltung
- Integration von Bildern; gleichzeitige Darstellung von Text und Bildern ist möglich
- Nicht über die Tastatur erreichbare Sonderzeichen können mit der Maus aus einer Tafel gewählt werden
- Serienbriefe
- Maximal anzeigbarer Textausschnitt 20 Zeilen mal 74 Zeichen
- Ausführliche Dokumentation

An 1st Word Plus fällt zunächst einmal das Fehlen von Bugs, also von Programmierfehlern, auf. Das ist für ein neues Produkt keinesfalls selbstverständlich und hier wohl in erster Linie der überaus kritischen Abnahme seitens der Firma Atari zu verdanken, was das Erscheinen ja auch erheblich verzögerte.

Neu gegenüber dem Vorgänger 1st Word ist die Möglichkeit, den gerade bearbeiteten Text auf Diskette (oder Festplatte) zu sichern und dann, ohne erneutes Laden, gleich weiterarbeiten zu können.

Ein geübter Vielschreiber wird, auf die Dauer gesehen, mit 1st Word Plus einige Probleme haben: Das Abrufen der Funktionen mittels Maus und Abroll-Menüs ist zwar sehr übersichtlich und leicht zu erlernen, dauert dafür aber auch recht lange und wird, wenn man sich mit dem System erst einmal gut auskennt, schnell lästig. Die Möglichkeit, wenigstens die häufigsten Befehle auch über Tastenkombinationen ansprechen zu können, wird dann schmerzlich vermißt. Weiter können auch keine Tastaturma-



Abbildung 3: Signet von Signum

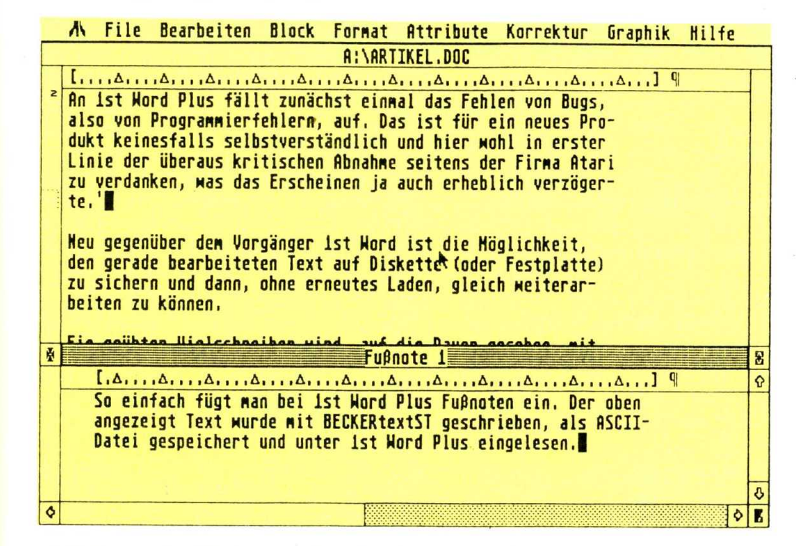


Abbildung 4: Bildschirmaufbau bei 1st Word Plus

kros definiert werden. Diese Möglichkeit, oft auch Floskeltasten genannt, würde es erlauben, längere Eingaben, Text und auch Steuerzeichen, selbst zu definieren und dann durch einfache Tastenkombinationen abzurufen. Die Konkurrenz ist da schon weiter.

BECKERtext ST

BECKERtext ST ist, grob gesagt, vergleichbar mit 1st Word-Plus. Da ein Test dieses Programms erst für eine der nächsten Ausgaben von ST vorgesehen ist, müssen wir an dieser Stelle etwas ausführlicher darauf eingehen. Zunächst beherrscht BECKERtext ST natürlich ebenfalls die Standardfunktionen eines Textsystems auf dem Atari:

- Volle Einbindung in GEM mit Darstellung von Schriftattributen auf dem Bildschirm
- Formatierbefehle zur Textdarstellung (Blocksatz, Zentrieren usw.)
- Blockoperationen
- Suchen/Ersetzen
- Individuelle Druckeranpassung ist möglich

Daneben werden folgende Besonderheiten geboten:

- Einfügen von Bildern in den Text
- Ausdruck des Textes in mehreren Spalten
- Befehle wahlweise mit der Maus oder über Tastenkombinationen erreichbar
- Eigene Tastaturmakros möglich (maximal 30 Makros zu je 160 Zeichen)
- Wörterbuch; wahlweise kann schon bei der Eingabe korrigiert werden
- automatische Trennung; die wenigen Fälle, in denen diese versagt, sind genau definiert und können über das Wörterbuch abgefangen oder von Hand nachkorrigiert werden
- Die Druckausgabe ist auch auf eine Datei möglich
- Mehrere Dateien können zusammenhängend gedruckt werden
- Inhaltsverzeichnis kann automatisch erstellt werden
- Sachregister (Index) kann automatisch erstellt werden
- Vom Programm eingefügte Leerstellen und vom Benutzer eingegebene RETURNs werden auf dem Bildschirm durch unaufdringliche Sondersymbole dargestellt.
- Im Text kann zeilen- und spaltenweise gerechnet werden
- Aktuelles Datum und Uhrzeit können BEIM DRUCKEN in den Text oder die Seitenüberschrift übernommen werden
- Es können Formulare erstellt werden, bei denen Eingabe nur an genau definierten Stellen möglich ist (Z. B. Rechnungen)
- Zeichen- und Zeilenabstand innerhalb des Textes veränderbar
- Wahlweise ST- oder PC-Zeichensatz; nicht über die Tastatur erreichbare Zeichen sind mit der Maus aus einer Tafel wählbar; bis zu 10 dieser Zeichen können auch auf die Sondertasten F1 bis F10 gelegt werden

- Editieren von C-Programmen wird durch wählbaren Sondermodus unterstützt
- Serienbriefe
- Maximal darstellbarer Textausschnitt 21 Zeilen mal 77 Zeichen oder 28 Zeilen mal 96 Zeichen wählbar
- Ausführliche und verständliche Anleitung mit Referenzkarte der wichtigsten Tastaturbefehle

Eine ordentliche Endkontrolle mußte BECKERtext ST anscheinend nicht über sich ergehen lassen, denn nur so lassen sich die vielen Unstimmigkeiten (darf ich's Fehler nennen?) erklären, die beim Betrieb auftreten. Zwar hat man bei DATA BECKER eine umgehende Beseitigung der gemeldeten Mängel zugesagt, aber in dem gegenwärtigen Zustand (April '87) kann man das Programm einem Anfänger eigentlich nicht zumuten. So wird etwa beim automatischen Einrücken links im Ausdruck (und NUR dort!) ein gleichgroßes Stück am rechten Rand weggelassen (wegen der Symmetrie?). Das für BECKERtext ST verantwortliche Programmierteam hat andererseits bereits dem Textomat seine Mängel gründlich ausgetrieben, und so besteht Grund zur Hoffnung. Für die Seriosität der Firma DATA BECKER spricht, daß die Updates für (legale!) Benutzer umsonst sind; daher kann man den Kauf von BECKERtext ST auch zum gegenwärtigen Zeitpunkt mit gutem Gewissen empfehlen.

Abgesehen von diesen kleinen Widrigkeiten ist das Arbeiten mit BECKERtext ST eine wahre Freude: Auf die Befehls- und Floskeltasten möchte man bald nicht mehr verzichten; die Kennzeichnung der vom Programm eingefügten Leerstellen und der RETURNs ist eine sehr nützliche Hilfestellung. Beim nachfolgenden Arbeiten mit 1st Word Plus fühlt man sich dann wie ein Autofahrer, dem plötzlich der Rückspiegel fehlt.

Das Erzeugen von reinem ASCII-Text ist bei BECKERtest ST ein echtes Problem. Richtig arbeiten kann man eigentlich nur im Textverarbeitungsmodus, der hier 'Brief-Modus' heißt. Gibt man hier nun reinen ASCII-Text ein und verzichtet man weiter auch auf sämtliche Textattribute, so wird dennoch überreichlich Steuerinformation mit dem Text abgespeichert. Das verwirrt selbst den abgehärtetsten Compiler. Dem ASCII-Modus von 1st Word Plus entspricht hier am ehesten der Modus 'C-Source', der aber für etwas anderes als für C-Quellprogramme praktisch nicht zu gebrauchen ist: Die Betriebsart 'Einfügen' läßt sich hier nicht abstellen, die Eingabe von geschweiften Klammern wird stets auch als Formatierbefehl verstanden und der Bildschirminhalt entsprechend gestylt. Geschweifte Klammern braucht man nicht? Dann arbeiten Sie mal mit TeX! Einzige Lösung: Den Text im 'Brief-Modus' erstellen, automatische Trennung abschalten, beim Abspeichern des Textes das Attribut ASCII auswählen. Dafür kann man das Lexikon bei Quellprogrammen, wie übrigens auch bei 1st Word-Plus, für eine einfache Syntaxprüfung einsetzen.

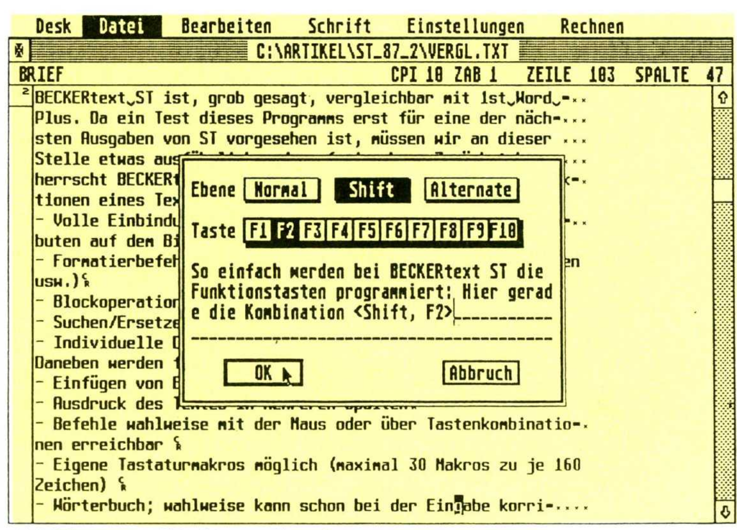


Abbildung 5: Bildschirmaufbau bei BECKERtext ST

Signum

Das System Signum wurde bereits in der ST 1/87 vorgestellt. Die Fähigkeiten von Signum beginnen dort, wo die gewöhnlicher Textsysteme wie 1st Word Plus und BECKERtext ST enden: bei der Eingabe von Formeln. Dies umfaßt auch alle grafischen Darstellungen, die sich aus kleinen Bausteinen der Größe von Buchstaben zusammensetzen lassen. Hier leistet Signum einfach Unglaubliches: Zeichen können auf dem Bildschirm in winzigen Stufen verschoben werden, die beim Ausdruck einer horizontalen Auflösung von 1/90 Zoll und einer vertikalen Auflösung von 1/54 Zoll entspricht. Neben vielen mitgelieferten Zeichensätzen (mittlerweile können auch ein kyrillischer und ein nicht-proportionaler Zeichensatz nachbestellt werden) wird auch ein Zeichensatzeditor mitgeliefert. Dieser erlaubt es, in einfacher Weise eigene Zeichen für Bildschirm und Drucker zu erstellen oder vorhandene nach eigenem Geschmack abzuändern. Bis zu sieben verschiedene Zeichensätze sind in einem Dokument gleichzeitig möglich.

Signum ist ein echtes WYSIWYG-System (sprich: wissiwig, von: What You See Is What You Get), bei dem bereits bei der Eingabe auf dem Bildschirm alles so angezeigt wird, wie es später gedruckt werden würde. Dazu gehören die eben angeführte feine Positionierung der Zeichen, die verschiedenen Zeichensätze und Proportionalschrift. Dies alles geht natürlich nur, wenn der bearbeitete Text nicht über den Zeichensatz des Atari, sondern mittels Grafik angezeigt wird. Neben den erstaunlichen Resultaten zeitigt dies aber noch einige weitere Nebeneffekte: Die Bildschirmdarstellung wird sehr langsam; auch ein ungeübter Tipper kann das System dazu bringen, die Zeichen mit deutlicher Verzögerung anzuzeigen. Zum anderen wird die abzuspeichernde Information natürlich recht umfangreich; man rechnet mit etwa 5 KByte pro Seite.

Ein derart leistungsfähiges grafisches System bietet selbstverständlich auch leistungsfähige Hilfsmittel: Frei defi-

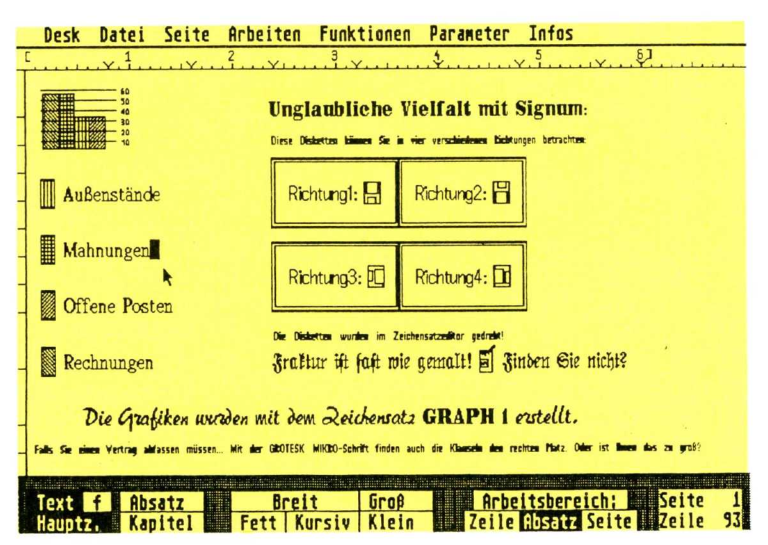


Abbildung 6: Bildschirmaufbau bei Signum

nierbare Tastaturmakros und Anzeige selbst des Textes am Kopf- und Fußende einer Seite gehören dazu.

Bei soviel Licht gibt es allerdings auch Schatten: Die üblichen Möglichkeiten eines Textsystems, beispielsweise Blockoperationen, sind deutlich zu kurz gekommen. Auch Serienbriefe sind nicht möglich, von dem Einbau von Bildern ganz zu schweigen.

Man kann dies alles aber auch ganz anders sehen: Signum ist nicht in erster Linie zur Bearbeitung von Texten, sondern für Formeln gedacht. Diese können direkt am Bildschirm selbst von Ungeübten nach kurzer Einarbeitungszeit meisterhaft gestaltet werden. Man sieht eben sofort, was dabei herauskommt und kann notfalls gleich korrigieren. Umfangreicher ASCII-Text kann zuvor mit einem der üblichen Textsysteme editiert und dann in ein Signum-Dokument übernommen werden. Was jedoch einmal in einem Signum-Dokument steht, ist für andere Textprozessoren unwiderbringlich verloren. Serienbriefe sind alleine schon wegen des erheblichen Druckaufwands mit Signum wenig sinnvoll; hier kann sich allerdings beim (zukünftig möglichen?) Einsatz eines Laserdruckers einiges ändern. Und was die Grafik angeht: Da sind seit der Weihnachtsfeier der Firma Application Systems Ende letzten Jahres die zuständigen Programmierer bereits dabei, die hauseigenen Produkte Signum und STAD (Test in ST 2/87) miteinander zu verheiraten. Wenn man die beiden Programmierer nach ihren Produkten beurteilt, wird man hier auf Erfolg wetten.

Einsatzgebiete

Fassen wir also das über die Einsatzgebiete Herausgefundene zusammen: 1st Word Plus und BECKERtext ST eignen sich sehr gut für die übliche Textverarbeitung; der gebotene Leistungsumfang hat hier bereits einen sehr hohen Standard erreicht. Dagegen ist Signum gefragt, wenn es einmal schwieriger wird: Die Bearbeitung von Formeln geht damit sehr leicht von der Hand; das gewohnte Prinzip, alles auf dem Bildschirm so zu sehen, wie es später gedruckt wird, ist hier zum Wohle des Benutzers vollständig durchgehalten.

1st Word plus und BECKERtext ST auf der einen und Signum auf der anderen Seite sind in diesem Sinne keine Konkurrenten: Sie ergänzen sich zwar gegenseitig, können sich aber nicht ersetzen.

Die Wahl zwischen 1st Word Plus und BECKERtext ST ist dagegen viel schwieriger: Zwar sprechen beide Systeme die gleiche Zielgruppe an und bieten auch vergleichbare Leistungen im Bereich der Grundaufgaben, bei den Feinheiten unterscheiden sie sich jedoch erheblich. Die bei 1st Word plus mögliche Fußnotenverwaltung wird manchen Benutzern die Wahl ohnehin schon abnehmen, andererseits ist BECKERtext ST mit mehrspaltigem Ausdruck, automatischem Inhalts- und Stichwortverzeichnis, Befehlstasten und Tastaturmakros sowie der Möglichkeit, die Druckausgabe auf Datei umzuleiten, eindeutig das professionellere System. Wenn nur die kleinen Fehler nicht (mehr) wären!

Dokumentation

An die Dokumentation sind, gerade wenn das System auch für Anfänger geeignet sein soll, besondere Anforderungen zu stellen. Eine ausführliche, gut verständliche und gegliederte Anleitung ist hier unabdingbare Voraussetzung: Ein Neuling muß alles im Handbuch finden und sich dann darauf verlassen können, daß auch alles hundertprozentig so funktioniert, wie es dort beschrieben steht. Diesem hohen Anspruch genügt keines der vorgestellten Systeme.

Das Handbuch von 1st Word Plus ist ein Spiralheft im Format DIN A5. Es ist ausführlich und vollständig, aber dennoch dringend verbesserungsbedürftig. Auf Seite 7 wird vom Leser gefordert, daß er mit den Grundfunktionen des Betriebssystems bereits umgehen kann. Dies ist durchaus sinnvoll. Auf Seite 8 wird dann diesem erfahrenen Leser erklärt, wie er das Gerät einzuschalten hat. Die Krönung ist eine vollständige Hardcopy des leeren Bildschirms auf Seite 8 unten. Wird es dann wirklich schwieriger, so hat der Ungeübte Mühe, den Text beim ersten Durchlesen überhaupt zu verstehen; vieles ist mißverständlich formuliert. Ist man aber erst einmal in das System eingearbeitet, dann hat man ein leistungsfähiges Nachschlagewerk. Die winzige Schrift des Handbauchs ist allerdings eine Zumutung. Bei vielen Bildschirmausdrucken wurde nur ein Teil des Bildschirms wiedergegeben, das vollständige Bild wäre oft sinnvoller. Ein Stichwortverzeichnis fehlt.

Bei BECKERtext ST wurde mit dem Verfassen des Handbuchs ganz bewußt jemand beauftragt, der nicht an der Entwicklung des Programms beteiligt war. Das Ergebnis bestätigt dieses Konzept aufs Eindrucksvollste. Neben einem gut gegliederten Inhaltsverzeichnis gibt es noch ein Stichwortverzeichnis und sogar ein Glossar. Das Handbuch kommt in einem Ringbuch des Formats DIN A5, man kann daher noch Notizen auf eigenen Blättern einlegen. Bei den aussagekräftigen Bild-

schirmausdrucken ist stets das gesamte Bild wiedergegeben; die Schrift ist groß und gut lesbar. Die Formulierung ist stets klar und auf Anhieb verständlich. Das beste Handbuch ist jedoch an den Stellen zum Scheitern verurteilt, an denen das Programm nicht so funktioniert, wie es sollte. Im Gegensatz zu 1st Word Plus und, wie wir gleich sehen werden, auch zu Signum muß bei BECKERtext ST nicht die Dokumentation, sondern das Programm verbessert werden.

Das Äußere des Handbuchs zu Signum ist ebenfalls sehr ansprechend. Auch hier handelt es sich um ein Ringbuch im Format DIN A5. Der Inhalt ist jedoch eine maßlose Enttäuschung. Der Umfang erreicht hier bei weitem nicht den bei 1st Word Plus oder BECKERtext ST. Und dieser äußere Schein trügt nicht: Es steht tatsächlich auch weniger drin. Dabei hat selbst der Profi bei Signum viel mehr zu lernen als bei den beiden anderen Systemen. Nicht, daß das Handbauch etwa unverständlich oder wortkart geraten wäre, nein, man kann sogar wie in einem Roman darin lesen und alles gleich am Bildschirm ausprobieren. Aber wehe, wenn man einmal etwas nachschlagen möchte oder gar etwas sucht, was nicht in aller Breite behandelt wird: Das Inhaltsverzeichnis ist nicht nur witzig formuliert, sondern in der Tat ein Witz, ein Stichwortverzeichnis fehlt, und mangels Strukturierung ist die gesuchte Information im Text selbst dann nur mit Mühe zu finden, wenn man schon ungefähr weiß, wo das Gesuchte steht. Und was nicht ausführlichst behandelt wurde, steht in der Regel überhaupt nicht drin. Umgekehrt kann man dies aber auch als Herausforderung betrachten: Als erste Übung mit Signum erstelle man ein Inhaltsverzeichnis, welches wenigstens die Abschnittsüberschriften enthält. Die Schrift ist auch hier zu klein geraten und durch das Druckverfahren zudem noch von minderer Qualität. Die Originalvorlage wurde mit Signum erstellt, sodaß man wenigstens eine gute Demonstration der erreichbaren Effekte erhält.

Wörterbuch und Trennung

Bei 1st Word Plus haben Wörterbuch und Trennung nichts miteinander zu tun. Die Trennung geschieht nicht schon beim Eingeben, sondern erst beim nachträglichen Formatieren halbautomatisch mit Hilfe einer Dialogbox. Es ist schon recht lästig, daß man sich um jede Trennung selbst kümmern muß.

Das Wörterbuch kann keine zusammengesetzten Wörter erkennen und wird dadurch recht umfangreich. Für jedes ins Wörterbuch neu aufzunehmende Wort muß mit der Maus eine Menü aufgerollt und dann ein Menüpunkt ausgewählt werden. Das ist eine üble Strafarbeit.

In BECKERtext ST sind Trennung und Wörterbuch miteinander verknüpft. Die automatische Trennung, die gleich bei der Eingabe geschieht, kann dort, wo sie tatsächlich einmal versagt, über das Wörterbuch korrigiert werden. Dieses nimmt nämlich, der Duden macht's vor, die korrekte Trennung gleich mit auf. Das Arbeiten geht einem auch hierbei viel leichter von der Hand als mit 1st Word Plus. Zusammengesetzte Wörter werden erkannt, Vorsilben können ebenfalls als selbständige Wörter abgespeichert werden. Der Umfang hält sich somit in Grenzen. Die leichte Bedienbarkeit ist hier allerdings auch unbedingt erforderlich: Das mitgelieferte Wörterbuch ist derart miserabel und erzeugt solch haarsträubende Trennfehler, daß man es am besten gleich wegwirft und ein eigenes aufbaut.

Signum besitzt kein Wörterbuch, und die Trennung ist hier am wenigsten komfortabel. Die zu trennende Stelle muß der Benutzer selbst markieren und dabei mit der Maus recht gut zielen.

Druckeranpassung

Bei der Druckeranpassung hat eindeutig 1st Word Plus die Nase vorn. Hier wird nicht nur das wahlfreie Umschalten auf Briefqualität, sondern auch die Grafikausgabe auf jeden beliebigen Drucker unterstützt. Für alle anderen Programme ist dies vorbildlich. Die Druckausgabe wird automatisch gespoolt, dazu muß allerdings die gewählte Datei auf dem Massenspeicher verfügbar bleiben: Also kein Diskettenwechsel! Beim Arbeiten mit einer Festplatte bedeutet dies natürlich keine Einschränkung.

BECKERtext ST erlaubt es, wie 1st Word Plus, die durch Steuerzeichen im Text angesprochenen Schriftattribute jedem beliebigen Drucker anzupassen. Zum Umschalten auf Briefqualität sind jedoch die Druckerschalter zu bemühen, dies kann man aber noch hinnehmen. Völlig unverständlich ist hingegen, daß Grafik nur auf Epson- und kompatiblen Drucker, beispielsweise Star NL 10 mit Parallel-Interface, ausgegeben werden kann. Dieser Pfusch wird hoffentlich in einer der nächsten Revisionen behoben sein. Im Hinblick auf semi-professionelle Druckausgabe mit 24-Nadel-Druckern ist es unverständlich, wie so etwas bei DATA BECKER durchgehen konnte. Nebenbei bemerkt ist dies bereits der zweite Sündenfall in dieser Richtung: Auch Text Design ST läuft nur mit Epson-kompatiblen Druckern und dies, obwohl der Profi Painter aus dem gleichen Hause vormacht, wie einfach hier eine Druckeranpassung ist.

Bei Signum entfällt das Thema Druckeranpassung: Ein Druckprogramm für 24-Nadel-Drucker und eines für 9-Nadel-Drucker werden mitgeliefert. Die Programme sind nicht zu modifizieren. Die Lieferfirma empfiehlt den NEC P6; das Programm für 9-Nadel-Drucker arbeitet nur mit Epson-kompatiblen.

Grafik im Text

1st Word Plus und BECKERtext ST gestatten das Einbinden von Bildern in den Text, bei Signum ist dies (noch) nicht möglich. Allerdings kann nur bei 1st Word Plus auch die Grafik zusammen mit dem Texxt auf dem Bildschirm, wenn auch mit einem deutlich kleineren Zeichensatz, dargestellt werden. Keines der beiden Systeme ist anscheinend in der Lage, Grafikausgabe eines der verbreiteten Zeichenprogramme zu übernehmen, bei beiden muß mittels eines Hilfsprogramms ein Teil des Bildschirms als Grafik in einer Datei abgespeichert werden. Nur die mit diesen speziellen Programmen produzierten Bilder können die Textsysteme dann verarbeiten. BECKERtext ST ist dabei ein klein wenig vielseitiger: Mit „seinem“ Hilfsprogramm kann man auch Dialog- und Alarmboxen einfangen.

Speicherformat

Alle drei Programme unterscheiden sich gründlich darin, wie formatierter Text abgespeichert wird. Hält 1st Word plus die gespeicherte Datenmenge nach allen Regeln der Kunst gering, so ist dieses Unterfangen bei Signum wegen der Fülle der zu speichernden In-

Hardware-Voraussetzungen

1st Word Plus:
- Atari ST mit 1 MB oder TOS in ROM
- Bildschirm: Monochrom oder Farbe
- Floppy
- Drucker: beliebig (Grafik nur auf Nadeldruckern)

BECKERtext ST:
- Atari ST mit TOS in ROM
- Bildschirm: Monochrom oder Farbe
- Floppy
- Drucker: beliebig (Grafik aber nur auf Epson-kompatiblen)

Signum:
- Atari ST (möglichst 1 MB und TOS in ROM)
- Bildschirm: Nur monochrom
- Floppy (möglichst 720 KB, also SF 314)
- Drucker: 24-Nadel-Drucker oder Epson-kompatibler 9-Nadel-Drucker
- Sinnvolles Zubehör: Festplatte

Die Preise

1st Word Plus	: 199 Mark (Atari, Raunheim)
BECKERtext ST	: 199 Mark (DATA BECKER, Düsseldorf)
Signum	: 445 Mark (Application Systems, Heidelberg)

formation von vorne herein zum Scheitern verurteilt. Durchschnittlich 5 KByte pro Seite sind der Preis für die äußerst flexible Textgestaltung. Wieder einmal völlig unverständlich ist allerdings das Konzept von BECKERtext ST. Hier wird jede Zeile grundsätzlich mit Bytes des Wertes Null bis zur eingestellten Zeilenlänge aufgefüllt. Jeder Informatikstudent im ersten Semester wäre bereits über eine solche Schlamperei erhaben. Hier ist, wie auch an anderen Stellen bei BECKERtext ST, Nachbesserung unbedingt erforderlich.

Kopierschutz

1st Word Plus ist nicht kopiergeschützt, hier wird auf die Ehrlichkeit der Kundschaft vertraut. Neue Produktinformationen und den vollen Service gibt es natürlich nur für legale Benutzer.

Bei BECKERtext ST hat man den von DATA BECKER bisher beschrittenen Irrweg endlich verlassen: Beim Programmstart muß nun nicht mehr jedesmal die Originaldiskette in Laufwerk A liegen. Dafür ist das Programm vor dem ersten Einsatz auf der Originaldiskette erst einmal zu initialisieren. Dabei trägt der Benutzer seinen Namen und Adresse ein und legitimiert sich dadurch für die bei BECKERtext ST noch dringend nötigen Updates. Das Programm selbst kann dann beliebig, auch auf Festplatte, verwendet werden, trägt aber den Namen des Käufers immer mit sich. Im Interesse der ehrlichen Kunden ist dies eine sehr faire Lösung.

Signum treibt den Kopierschutz am weitesten. Nach dem Einschalten des Rechners und vor dem ersten Aufruf von Signum ist das mitgelieferte Progarmm INSTAL.PRG zu starten, das eine Betriebssystem-Erweiterung installiert. Dabei muß eine der beiden im Lieferumfang enthaltenen Systemdisketten im Laufwerk A liegen. Dies ist recht umständlich; da man die Originaldiskette bei weiteren Programmaufrufen aber nicht mehr braucht, hält sich der Aufwand immerhin in Grenzen.

Kaufentscheidung: Stufenkonzept

Für anspruchsvolle Textverarbeitung ergibt sich somit, wenn sie nach dem WYSIWYG-Prinzip funktionieren soll, folgende Empfehlung: Ein Textprozessor herkömmlicher Art, beispielsweise 1st Word Plus oder BECKERtext ST, wird für reine Textverarbeitung auf jeden Fall benötigt und ist dafür auch am besten geeignet. Geht es um Formeln, besondere Schriftarten oder Proportionaldruck, so ist hierfür unbedingt ein im Hinblick auf diese Anwendung speziell entwickeltes System, wie etwa Signum, zusätzlich einzusetzen. Dafür ist Signum zur reinen Textverarbeitung weniger geeignet. Die Entscheidung zwischen 1st Word Plus und BECKERtext ST ist sicher nicht ganz einfach; hier gibt es allerdings auch noch andere Bewerber.

Dr. V. Kurz

ENDLICH! Start der Reihe EDV BUCHFÜHRUNG LEICHTGEMACHT



Sind Sie Selbständiger, Gewerbetreibender oder Steuerberater, Buchführungs-Neuling oder routinierter Profi,
können Sie sich keine lange Einarbeitungszeit in umständliche Programme leisten,
dann brauchen Sie
eine **dialogorientierte, menügesteuerte** Software mit übersichtlichen Eingabebildschirmen, für Sie von Praktikern geschrieben,
mit der Sie endlich

BUCHEN OHNE STRESS



- läuft auf jeder gängigen Atari ST Hardware Konfiguration,, **natürlich unter GEM, kein Kopierschutz.** Jede Diskette enthält lediglich eine Seriennummer zur Rückverfolgung unerlaubter Kopien. Eine eventuell vorhandene Hardwareuhr kann selbstverständlich weiter benutzt werden. **Optimaler Anwendungskomfort** ist für **fibuMAN** oberstes Gebot
- gibt Ihnen in Sekundenschnelle zu **jedem** Zeitpunkt einen präzisen Überblick über Ihre Geschäftsfinanzen
- hält **alle Listen in chronologischer Ordnung** bereit
- integriertes Abschreibungsprogramm

- **weist selbsttätig auf unsinnige Buchungen hin.** Der Buchungsneuling lernt bequem während er bucht!
- entscheidet für Sie, ob MWST oder Vorsteuer zu buchen sind
- rechnet die MWST von **Nettorechnungen** selbst hoch
- erlaubt schnelles Suchen nach bestimmten Buchungen oder verschiedenen Kriterien in allen relevanten Listen – **sogar mit sachgerechter Übernahme in die aktuelle Buchung** (z. B. Debitoren, Kreditoren Ausbuchungen, offen Posten!
- **storniert JEDEN Buchungssatz des Journals auf Tastendruck oder Mausklick!**
- läßt Sie auch während des Buchungsvorgangs neue Konten anlegen und gibt Ihnen dabei Auskunft über die Bedeutung dieses Kontos für die verschiedenen Auswertungen (mit Beispielen!)
- informiert Sie vor und nach jeder Buchung über die Kontostände der gerade angesprochenen Konten
- ordnet **jedem Konto einen optionalen Automatik-Text** und einen bestimmten MWST Satz zu (kann jederzeit überschrieben werden)
- arbeitet mit oder ohne Belegnummer, je nachdem wie Sie es gewohnt sind
- **druckt die Umsatzsteuer Voranmeldung mit allen Daten direkt auf das Formular des Finanzamts** (nur unterschreiben müssen Sie noch selbst!)
- druckt auf Abruf ALLE vom Finanzamt zum Abschluß vorgeschriebenen Auswertungen hintereinander (beliebige Anzahl von Kopien!)
- **bucht** nach einmaliger Eingabe **automatisch** alle **sich regelmäßig wiederholenden Vorfälle**
- kümmert sich um Sicherheitskopien und Datenschutz (frei definierbares Kenntwort)
- zwingt Sie, das Systemdatum zu stellen, aber nur, wenn Sie es vergessen haben sollten
- verfügt über eine Schnittstelle zu einer extrem komfortablen Faktura (**faktuMAN** in Vorbereitung), getreu der **fibuMAN** Devise:
  EIN VORGANG – EINE EINGABE – EINE WIRKUNG
- ist für Sie ungeeignet wenn Sie mehr als **64 000 Buchungen im Monat** haben oder **99 999 Konten** nicht ausreichen sollten

fibuMAN E Einnahmen-Überschuß Rechnung

einfach und schnell zu erlernen mit allen oben genannten Vorzügen, Auswertungen u. a. Saldenliste, Finanzkonten, Umsatzsteuervoranmeldung, Einnahmen-Überschuß Rechnung
incl. ausführlichem Handbuch **DM 345,–**

fibuMAN F Finanzbuchhaltung

wie **fibuMAN E**, jedoch keine Einnahmen-Überschuß Rechnung, statt dessen Gewinn & Verlustrechnung und Bilanzierung **nach den neuesten gesetzlichen Vorschriften**, ermöglicht sowohl Debitoren-Kreditoren als auch Offene Posten Buchhaltung. **DM 645,–**

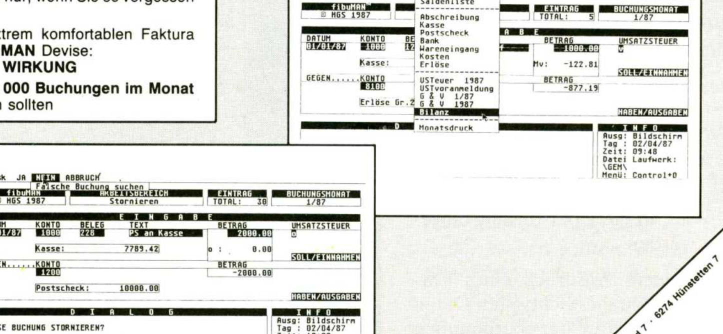
fibuMAN M Mandantenfähige Finanzbuchhaltung

vereinigt alle Eigenschaften von **fibuMAN E** und **fibuMAN F**.
Ideal für Steuerberater und Inhaber mehrerer Firmen, die sowohl Einnahmen-Überschuß Rechnung, als auch Bilanzierung benötigen und beides in gleicher komfortabler Weise bedienen wollen. Mandanten oder Firmen vom Programm aus mit allen zugehörigen Stammdaten zu verwalten. **DM 845,–**

Individuelle Anpassung des Systems nach Absprache, Hotline, Update Service, und und und...

Ein Aufstieg in der **fibuMAN** Reihe ist jederzeit unter Anrechnung des aktuellen Preises der ursprünglichen Version möglich.

Demodisk mit Handbuch (wird beim Kauf angerechnet) **DM 60,–**





MegaSoft · Tel. 0 64 38 - 44 63
Auf der Hosterd 7 · 6274 Hünstetten 7

Bestell-Coupon Einsenden an: MegaSoft · Auf der Hosterd 7 · 6274 Hünstetten 7

Bitte liefern Sie:

___ fibuMan E (Einn.-Übersch. Rechnung) 345,– DM
___ fibuMan F (Finanzbuchhaltung) 645,– DM
___ fibuMan M (Finanzbuchh. Mandantenf.) 845,– DM
___ fibuMan Handbuch (Anrechnung bei Kauf eines Paketes) 60,– DM

☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Je Lieferung zuzügl. 5,– DM Versandkostenpauschale

Name: ______
Straße: ______
Ort: ______

„Weisung", ein Programm zum Drucken von Überweisungsformularen:

Die Idee für das Programm entstand, nachdem ich etliche Überweisungen mühsam geschrieben habe. Sogleich habe ich mich mit den Anforderungen für ein solches Programm auseinandergesetzt und kam zu folgenden Leistungsmerkmalen:

- Einfaches Editieren des Überweisungsformulars am Bildschirm;
- Vorbereitete Formulare sollen eingeladen und zu jedem Zeitpunkt ergänzt werden können;
- Speichern der Formulare auf Diskette;
- Laden der Überweisungsformulare von Diskette;
- Verwendung von Datensätzen, die aus einer Datenbank kommen (jedes Formular – ein Datensatz – eine Datei);
- Drucken des Formulars auf übliches Endlosüberweisungspapier;
- Der Ausdruck von einheitlichen Überweisungsvordrucken soll möglich sein;
- Diese Schritte müssen zu jedem Zeitpunkt möglich sein, unabhängig davon ob die Überweisung vollständig ausgefüllt ist;
- Das Computerdatum soll, wenn gewünscht, in die Überweisung eingesetzt werden;
- Nach jedem Ausdruck eines Formulars sollen die wichtigsten Daten in eine Protokolldatei übernommen werden und mit z.B. 1st-Word oder einer Datenbank weiterverarbeitet werden können;
- Das Programm soll voll GEM-unterstützt arbeiten.

Abb. 1: Hardcoppy von Eingabe-Menü

Arbeitsmaterial:

Ich habe das Programm mit dem Megamax Compiler und dem Megamax Construction-Set erstellt. Die Gerätekonfiguration des Atari ST spielt keine Rolle, jedoch kann es bei den Druckern Schwierigkeiten geben, da sie unterschiedliche Steuercodes haben. Dieses Problem wird aber mittels einer Druckeranpassung komfortabel gelöst. Es ist wünschenswert, daß der Drucker einen Traktor für Endlos-

Überweisungsauftrag an 577 615 91 Buchungsvermerke der Bank

Volksbank Bad Neuenahr-Ahrweiler eG
5483 Bad Neuenahr-Ahrweiler

Empfänger (Name und Anschrift): Egon Müller, Mauergasse 32 / 8000 München 3
Bankleitzahl: 12 345 678
Konto-Nr. des Empfängers: 100 456 78
bei – (Volksbanken, Raiffeisenbanken usw.) oder ein anderes Konto des Empfängers*): Kreditbank des Volkes
Verwendungszweck (nur für Empfänger): Autokauf
DM: **8000,00**
Konto-Nr. des Auftraggebers: 2000 234 567
Auftraggeber: Karl-Heinz Kleinschmidt, Bachstraße 123 / 5000 Köln 41

*) Soll die Überweisung auf ein anderes Konto ausgeschlossen sein, so sind die Worte „oder ein anderes Konto des Empfängers" zu streichen.

06.04.1987
Datum Unterschrift

422 002 dgverlag 9

Mehrzweckfeld | Konto-Nr. | Betrag | Bankleitzahl | Text

Bitte dieses Feld nicht beschriften und nicht bestempeln

Überweisungsauftrag an 577 615 91 Buchungsvermerke der Bank

Volksbank Bad Neuenahr-Ahrweiler eG
5483 Bad Neuenahr-Ahrweiler

Empfänger (Name und Anschrift): Tanzschule WURMI / 5483 Bad Neuenahr-Ahrweiler
Bankleitzahl: 577 615 91
Konto-Nr. des Empfängers: 123 456
bei – (Volksbanken, Raiffeisenbanken usw.) oder ein anderes Konto des Empfängers*): Volksbank Bad Neuenahr-Ahrweiler EG
Verwendungszweck (nur für Empfänger):
NAME : ORT :
VORNAME : PLATZ:
TANZKURS: HONORAR p PERS:
DM:
Konto-Nr. des Auftraggebers: Auftraggeber:

*) Soll die Überweisung auf ein anderes Konto ausgeschlossen sein, so sind die Worte „oder ein anderes Konto des Empfängers" zu streichen.

Datum Unterschrift

422 002 dgverlag 9

Mehrzweckfeld | Konto-Nr. | Betrag | Bankleitzahl | Text

Bitte dieses Feld nicht beschriften und nicht bestempeln

Überweisungsauftrag an 577 615 91 Buchungsvermerke der Bank

Volksbank Bad Neuenahr-Ahrweiler eG
5483 Bad Neuenahr-Ahrweiler

Empfänger (Name und Anschrift): Klaus Mayer, Platanenweg 12 / 5400 Koblenz 1
Bankleitzahl: 101 11 101
Konto-Nr. des Empfängers: 321 098
bei – (Volksbanken, Raiffeisenbanken usw.) oder ein anderes Konto des Empfängers*): Bank für Kreditwesen
Verwendungszweck (nur für Empfänger): bekannt
DM: **1234,50**
Konto-Nr. des Auftraggebers: 315 894 23
Auftraggeber: Sabine Fischer, Hauptstraße 7a / 1625 Kleinkleckersdorf

*) Soll die Überweisung auf ein anderes Konto ausgeschlossen sein, so sind die Worte „oder ein anderes Konto des Empfängers" zu streichen.

24.11.1984
Datum Unterschrift

422 002 dgverlag 9

Mehrzweckfeld | Konto-Nr. | Betrag | Bankleitzahl | Text

Bitte dieses Feld nicht beschriften und nicht bestempeln

Abb. 2: Abgedruckte Überweisung

papier besitzt, da die Überweisungsformulare der Banken als Endlospapier erhältlich sind. Das abgedruckte Programm läuft auf dem Epson RX-80 einwandfrei und dürfte auch für alle anderen Epson-kompatiblen Drucker ohne Änderungen laufen.

Damit das Programm nicht zu umfangreich wurde, habe ich mich begnügt, mit einer großen Dialogbox zu arbeiten, die während des gesamten Programmablaufs auf dem Bildschirm steht. Diese Dialogbox enthält das Überweisungsformular, welches editiert werden kann und außerdem noch vier Knöpfe, die für die folgenden Funktionen „Überweisungladen", „-speichern", „-drucken", „Programm beenden" und „aktuelles Datum in die Überweisung eintragen", dienen. GEM übernimmt die Eingabe aller Felder der Überweisung, und erst wenn ein Knopf betätigt worden ist, übernimmt das Programm die Kontrolle. Hierdurch ist gewährleistet, daß zu jedem Zeitpunkt im Formular gearbeitet werden kann und nur bei der Ausführung der Funktion über die Knöpfe das Editieren nicht möglich ist. Hiermit ist die Funktionsweise des Programms bereits komplett erklärt.

Damit Sie beim Abtippen kein Programmfragment erstellen, welches nicht funktioniert, will ich einige Tips geben:

Es ist Ihnen freigestellt, Namen und Routinen umzuschreiben, doch möchte ich Ihnen empfehlen, erst einmal meine Vorschläge zu übernehmen, damit es keine Komplikationen zwischen den Namen, die Sie mit dem Construction-Set eingeben, gibt.

1. Erstellen Sie sich mit dem Construction-Set eine Dialogbox und geben Sie ihr den Namen „FORMULAR". Diese Dialogbox enthält eine weiße Box, die Sie mäglichst auf volle Größe ziehen sollten, damit genügend Platz vorhanden ist. Den dünnen Rand können Sie entfernen, und das Füllmuster kann frei gewählt werden.

2. Erstellen Sie sich vier Knöpfe mit mindestens dem Status EXIT, damit Sie die Dialogbox beim Anklicken der Knöpfe auch verlassen können. Geben Sie diesen Knöpfen mit der Namensfunktion des Construction-Sets folgende Namen: „LOAD", „SAVE", „PRINT", „QUIT" und „DATE". Die Beschriftung und die

Maße der Knöpfe können Sie frei wählen.

3. Als nächstes ist es empfehlenswert, eine weitere Box (das eigentliche Überweisungsformular) in die Dialogbox zu holen. Diese bekommt keinen Namen, da sie lediglich der Optik dient.
4. Nun können Sie zur visuellen Trennung der Felder acht kleinere Boxen auf die Überweisungsbox holen. Namen sind hier auch unerheblich.
5. Beschriften Sie die Boxen mit ‚TEXT', wobei Sie die Schriftgröße auf klein setzen (‚SM').
6. Nun kommen die eigentlichen Eingabefehler, die Sie sich in die Dialogbox holen und auf die entsprechenden Längen ändern. Danach geben Sie den einzelnen Feldern die Namen, die hier in einer kurzen Liste zusammengefaßt worden sind:

(Siehe Abbildung 3)

Sehen Sie sich die Hardcopy des Formulars an, in dem die Namen in den Feldern eingetragen sind. Wichtig ist beim Erstellen, daß Sie die Zeile mit den Festvorbesetzungen mit /~/ auf die entsprechende Länge auffüllen. Die Eingabemöglichkeiten sind mit ‚X' auszufüllen, da wir in jedem Feld alle Zeichen, Zahlen und Sonderzeichen zulassen wollen. In der Vorbesetzung sollten ‚_' eingegeben werden, da sonst Komplikationen auftreten.

7. Mit ‚STRING' können Sie eine Punktreihe für die Unterschrift erzeugen.
8. Speichern Sie die so erstellte Dialogbox unter dem Namen „WEISUNG" ab.
9. Tippen Sie das Programm möglichst fehlerfrei ab, kompilieren und linken es, und schon sind Sie fertig.

Haben Sie jetzt noch Schwierigkeiten mit der Reihenfolge, in der die Editfelder abgearbeitet werden, so können Sie die Felder austauschen, so daß der Cursor (Strich) beim Drücken von „CRSR UP", „SHIFT-TAB", „CRSR DOWN" und „TAB" in der gewünschten Reihenfolge springt. Dabei ist nicht nur der Name zu ändern, sondern das gesamte Feld muß verschoben und die entsprechenden Längen geändert werden. Dies liegt an der Struktur, die das Construction-Set beim Erstellen

Feld	Länge	Zweck
– „EMPF_1"	mit 40 Zeichen	für Name und Anschrift des
– „EMPF_2"	mit 40 Zeichen	Empfängers;
– „BLZ"	mit 13 Zeichen	für die Bankleitzahl;
– „KTO_EMPF"	mit 13 Zeichen	für die Kontonr. des Empfängers;
– „BANK"	mit 40 Zeichen	für das Bankinstitut des Empf.;
– „ZWECK_1"	mit 40 Zeichen	für den Verwendungszweck;
– „ZWECK_2"	mit 40 Zeichen	
– „ZWECK_3"	mit 40 Zeichen	
– „BETRAG"	mit 11 Zeichen	für den Überweisungsbetrag;
– „KTO_AUFT"	mit 13 Zeichen	für das Konto des Auftraggebers;
– „ANSCHR_1"	mit 40 Zeichen	für die Anschrift des Auftraggeb.;
– „ANSCHR_2"	mit 40 Zeichen	
– „DATUM"	mit 10 Zeichen	für das Datum.

Abbildung 3

```
 1:  /****************************************************************************/
 2:  /*                                                                          */
 3:  /*                 Druckprogramm für Überweisungsformulare                  */
 4:  /*                                                                          */
 5:  /*   programmiert von Ulrich und Holger Stumm im März 1987  (UHS-Software)  */
 6:  /*                                                                          */
 7:  /****************************************************************************/
 8:
 9:  int mo_hiden=0;                                  /* Steuerung des Mauszeiger */
10:  #define hidemaus() {if (mo_hiden==0) {graf_mouse(256,0L); mo_hiden=1;}}
11:  #define showmaus() {if (mo_hiden==1) {graf_mouse(257,0L); mo_hiden=0;}}
12:  #define pfeilmaus() graf_mouse(0,0L)              /* Mausform als Pfeil      */
13:  #define bienemaus() graf_mouse(2,0L)              /* Mausform als Biene      */
14:
15:  #include "gembind.h"
16:  #include "taddr.h"
17:  #include "obdefs.h"
18:  #include "osbind.h"
19:  #include "stdio.h"
20:  #include "weisung.h"  /* Datei muß erst mit Construction-Set erstellt werden */
21:
22:  #define tiefe 3                           /* Ebenen der Dialogbox 'formular' */
23:
24:  int contrl[12],intin[128],ptsin[128],ptsout[128],intout[128];
25:  int work_out[57],work_in[12];
26:
27:  int xd,yd,wd,hd;                                        /* Dialogboxgrößen */
28:
29:  int handle;                                       /* Arbeitsstationsnummer */
30:  FILE *file_handle;                                   /* für Dateiverwaltung */
31:
32:  long form_addr;                         /* Adresse des Dialogbaums 'FORMULAR' */
33:  long tree;                               /* Adresse für beliebigen Dialogbaum */
34:
35:  char pfad[64],                           /* Laufwerk + Suchpfad 'C:\MEGAMAX' */
36:       wahl[13],                           /* Auswahlkriterium      '\*.*'     */
37:       name[13],                           /* Filename der Ausgewählt wurde    */
38:       pfad_name[64];                      /* Laufwerk + Suchpfad + Filename   */
39:
40:  char protoname[64];                      /* Pfad und Namen für Protokolldatei */
41:
42:  char *text;                                        /* allgemiener Textpointer */
43:  char ***p_form[13];               /* Enthält Felder für ein Überweisungsformular */
44:
45:  int knopf,taste;                         /* für Rückmeldungen aus Funktionen */
46:  int xy[4];                        /* für v_bar (weißes Rechteck als Leiste ) */
47:
48:  /****************************************************************************/
49:
50:  gem_init()                 /* nötige Initialisierungen für den Betrieb mit GEM */
51:  {
52:    int i;
53:
54:    appl_init();                                        /* Anwendung anmelden */
55:    for (i=0;i<10;i++)
56:      work_in[i]=1;
57:    work_in[10]=2;
```

festlegt. Dies ist nämlich ein Baum mit einer Wurzel, und für jedes Objekt wird ein Knoten angelegt, der auf das nächste Objekt zeigt. Diese Reihenfolge können Sie nicht im Baum ändern, da – wenn der Baum einmal festgelegt ist – seine Struktur in den Grundzügen erhalten bleibt, sondern nur, indem Sie die Felder geschickt auf dem Bildschirm austauschen.

Nun dürften aber alle Schwierigkeiten beseitigt sein, und ich möchte Ihnen nun noch kurz die Funktionsweise der einzelnen Programmteile erklären:

main (), gem init ():

Der Programmablauf startet bei ‚main ()' und verzweigt direkt zu gem_init ()', wo einige nötige Initialisierungen vorgenommen werden. Außerdem wird das Resource-File nachgeladen und die Adresse der Dialogbox ermittelt. Wenn das File „WEIUNG.RSC" nicht gefunden worden ist, bricht das Programm mit einer Fehlermeldung ab. Da aber meist alles ohne Fehler abgelaufen ist, kehren wir in ‚main ()' zurück, um eine Statuszeile oben auf dem Bildschirm zu zeichnen. Anschließend werden Pointer auf die Editfelder gesetzt, damit wir sie vorbesetzen und auslesen können. Dann wird die Dialogbox ‚FORMULAR' auf dem Bildschirm unterhalb der Statuszeile gezeichnet. GEM übernimmt nun die Verwaltung der Knöpfe und der Editfelder mit ‚form_do'. Sobald ein Knopf gedrückt wird, gibt GEM die Kontrolle an unser Programm ab, und wir verzweigen – je nachdem welcher Knopf gedrückt worden ist – in die Unterroutinen.

Wird der Knopf namens ‚QUIT' angeklickt, so wird nachgefragt, ob man das Programm auch wirklich verlassen möchte. Beim Verlassen des Programms wird die Dialogbox geschlossen und die Arbeitsstation abgemeldet.

laden ():

In der Prozedur ‚laden ()' wird die Funktion ‚file_box (...)' aufgerufen. Diese habe ich selbst geschrieben, damit die Handhabung der Box um einiges leichter wird. Die Routine sorgt dafür, daß der Pfad, in dem man zuletzt war, bei erneutem Aufruf wiederbenutzt wird. Außerdem wird ein selektierter Filename auch wieder angezeigt. Die Funktion gibt den fertigen Filenamen zum Laden zurück. Wird

```
58:     v_opnvwk(work_in,&handle,work_out);                    /* Arbeitsstation öffnen */
59:     if (rsrc_load("WEISUNG.RSC")==0)                              /* RSC-File laden */
60:     {
61:       text="[3][Achtung Fehler| |Das RSC-File ist nicht vorhanden.][Abbruch]";
62:       form_alert(1,text);
63:       gemdos(0x0); /* Abbruch */
64:     }
65:     rsrc_gaddr(0,FORMULAR,&form_addr);         /* Adresse für Dialogbaum ermitteln */
66:   }
67:
68:   /************************************************************************/
69:
70:   laden()                             /* Überweisungsformular von Diskette laden */
71:   {
72:     int fehler,i,k;
73:     char zeile[255];
74:
75:     taste=file_box(pfad,wahl,name,pfad_name);                 /* Dateinamen auswählen */
76:     objc_draw(form_addr,5,tiefe,0,19,639,399);           /* Bildschirm restaurieren */
77:     if (taste==1)
78:     {
79:       bienemaus();
80:       if ((file_handle=fopen(pfad_name,"r"))!=0)           /* Datei zum Lesen öffen */
81:       {
82:         i=0;
83:         while((fehler!=0)&&(i<=12))               /* 12 Zeilen sollen gelesen werden */
84:         {
85:           k=0;
86:           do
87:           {
88:             fehler=(int)(fgets(&zeile[k],2,file_handle)); /* ein Zeichen lesen */
89:           }
90:           while((fehler!=0)&&(zeile[k++]!='\n'));
91:           zeile[k-1]='\0';
92:           sprintf(**p_form[i++],"%s",zeile); /* Textzeile in Editfeld eintragen*/
93:         }
94:         if ((fehler==0)&&(i!=13))
95:           form_alert(1,"[3][Die Formulardatei|ist fehlerhaft][Abbruch]");
96:         fclose(file_handle);                                      /* Datei schleißen */
97:       }
98:       else                                       /* wenn Datei nicht vorhanden ist */
99:         form_alert(1,
100:                "[3][Die Formulardatei konnte|nicht geöffnet werden][Abbruch]");
101:       pfeilmaus();
102:     }                                                                /* end of if */
103:   }                                                          /* end of prozedure */
104:
105:   /************************************************************************/
106:
107:   speichern()             /* speichern eines Überweisungsformulars auf Diskette */
108:   {
109:     int i;
110:
111:     taste=file_box(pfad,wahl,name,pfad_name);                 /* Dateinamen auswählen */
112:     objc_draw(form_addr,5,tiefe,0,19,639,399);           /* Bildschirm restaurieren */
113:     if (taste==1)                          /* wenn Fileselectbox mit  "OK" beendet */
114:     {
115:       bienemaus();
116:       file_handle=fopen(pfad_name,"w");                  /* Datei zum Schreiben öffen */
117:       for (i=0; i<=12; i++)
118:         fprintf(file_handle,"%s\n",**p_form[i]);  /* Editfelder rausschreiben  */
119:       fclose(file_handle);                                        /* Datei schließen */
120:       pfeilmaus();
121:     }
122:   }
123:
124:   /************************************************************************/
125:
126:   drucken()               /* druckt Editfelder auf Überweisungs-Endlospapier */
127:   {
128:     int i;
129:     char zeile[81];                              /* nimmt immer eine Druckzeile auf */
130:
131:     do                                             /* drucken ? und Drucker einschalten */
132:       taste=form_alert(1,
133:             "[3][Überweisung drucken?|Drucker einschalten!][ drucken | Abbruch ]");
```

```
134:      while((!Cprnos())&&(taste==1));              /* ist der Drucker eingeschaltet */
135:
136:      if (taste==1)                           /* wenn mit 'drucken' beendet worden ist */
137:      {                                            /* Schriftart für Drucker wählen */
138:        taste=form_alert(1,
139:                "[1][Wählen Sie eine Schriftstärke][ NORMAL | DICK | SUPER ]");
140:        bienemaus();
141:        switch(taste)
142:        {
143:          case 1:                                                   /* Doppeldruck */
144:               print("\033G");
145:               break;
146:          case 2:                                                   /* Fettschrift */
147:               print("\033E");
148:               break;
149:          case 3:                                     /* Doppeldruck + Fettschrift */
150:               print("\033G");
151:               print("\033E");
152:               break;
153:        }
154:        print("\033R\002");                    /* ESC "R" 2 (Zeichensatz DEUTSCH) */
155:        print("\033A\006");              /* ESC "A" 6 (Zeilenabstand auf 6/72 inch) */
156:
157:        sprintf(zeile," %s\n",**p_form[0]);
158:        print(zeile);                                         /* Empfänger 1. Zeile */
159:        sprintf(zeile,"%43s%s\n","",**p_form[2]);
160:        print(zeile);                                              /* Bankleitzahl */
161:        sprintf(zeile," %s\n\n\n\n",**p_form[1]);
162:        print(zeile);                                         /* Empfänger 2. Zeile */
163:        sprintf(zeile," %s",**p_form[3]);
164:        print(zeile);                                      /* Kontonummer Empfänger */
165:        sprintf(zeile,"\r%16s%s\n\n","",**p_form[4]);
166:        print(zeile);                                             /* Bankinstitut */
167:        sprintf(zeile,"\n\n%s\n\n",**p_form[5]);
168:        print(zeile);                                /* Verwendungszweck 1. Zeile */
169:        sprintf(zeile,"%s",**p_form[6]);
170:        print(zeile);                                /* Verwendungszweck 2. Zeile */
171:        sprintf(zeile,"%s",**p_form[8]);
172:        if (strlen(zeile)!=0)                         /* Nur wenn Betrag vorhanden */
173:          sprintf(zeile,"\r%41s**%s**\n\n","",**p_form[8]);
174:        else
175:          sprintf(zeile,"\n\n");
176:        print(zeile);                                   /* Überweisungsbetrag */
177:        sprintf(zeile,"%s\n\n\n\n",**p_form[7]);
178:        print(zeile);                                /* Verwendungszweck 3. Zeile */
179:        sprintf(zeile,"%15s%s\n","",**p_form[10]);
180:        print(zeile);                         /* 1. Zeile Anschrift Auftraggeber */
181:        sprintf(zeile," %s\n",**p_form[9]);
182:        print(zeile);                                /* Kontonummer des Empfänger */
183:        sprintf(zeile,"%15s%s\n\n\n\n\n","",**p_form[11]);
184:        print(zeile);                         /* 2. Zeile Anschrift Auftraggeber */
185:        sprintf(zeile,"%17s%s\n","",**p_form[12]);
186:        print(zeile);                                                     /* Datum */
187:        print("\0332");                             /* Zeilenabstand auf 1/6 inch */
188:        for (i=1; i<=12; i++)                      /* auf nächste Seite mit LF's */
189:          print("\n");
190:        print("\033F");                            /* Fettschrift ausschalten      */
191:        print("\033H");                            /* Doppeldruck ausschalten      */
192:        protokoll();                             /* Überweisung in Liste sichern */
193:        pfeilmaus();
194:      }
195:    }
196:
197:    /****************************************************************************/
198:
199:    print(string)                         /* konvertiert und druckt einen String aus */
200:    char string[255];
201:    {
202:      int i=0,flag=1;
203:
204:      while(string[i]!='\0')                  /* Solange das Ende nicht erreicht ist */
205:      {
206:        switch(string[i])                     /* eventuelle Konvertierung für Drucker */
207:        {
208:          case '[illegible]':  string[i]=64;   flag=1;  break; /* mit DEUTSCH-Zeichensatz */
209:          case 'Ä':  string[i]=91;   flag=1;  break;
```

die Filebox mit ‚Abbruch' beendet, so ist der Funktionswert 0. Nach dem Auswählen eines Formular-Files wird der Bildschirm durch erneutes Zeichnen der Dialogbox restauriert. Dies ist zwar nicht die feinste Methode, jedoch die einfachste. Nun wird die gewählte Datei zum Lesen geöffnet. Um eine Zeile in der Datei zu lesen, wird immer nur ein Zeichen gelesen un auf ‚/n' (Line Feed) geprüft. Dabei wird ein Zeilenstring aufgebaut, der anschließend ins entsprechende Editfeld geschrieben wird. Werden weniger als 12 Zeilen gelesen, wird die Datei als fehlerhaft erkannt. Wenn Sie in der Filebox einen Fantasienamen eingegeben haben, und diese Datei damit nicht vorhanden ist, wird ebenfalls eine Warnmeldung ausgegeben.

speichern ():

Die Prozedure ‚speichern ()' läuft in etwa so wie ‚laden ()' ab, nur daß hier für jedes Editfeld eine Zeile in die Datei geschrieben wird, die vorher zum Schreiben geöffnet worden ist.

drucken ():

In der Prozedure ‚drucken ()' kann man, falls man irrtümlich hierher geraten ist, in die Dialogbox zurückkehren. Außerdem kann man erst drucken, wenn der Drucker eingeschaltet ist. Die Funktion ‚Cprnos ()' liefert den Wert 1, wenn der Drucker angeschlossen und eingeschaltet ist. Danach kann man eine von drei Schriftarten wählen. Natürlich ist es Ihnen überlassen, zum Beispiel Doppeldruck in NLQ im Programm abzuändern. Jedoch müssen Sie in der darunterliegenden CASE-Anweisung die entsprechenden Steuercodes einsetzen. Nun wird der deutsche Zeichensatz im Drucker angewählt und der Zeilenvorschub auf die Hälfte reduziert. Ab jetzt werden nur noch halbe Zeilen pro Zeilenvorschub vorgeschoben, (normal 1/6 Zoll – halbe Weite 1/12 Zoll also 6/72 Zoll). In den nächsten 27 Zeilen werden die jeweiligen Druckzeilen zusammengesetzt und gedruckt. Dann wird der Zeilenabstand wieder auf 1/6 Zoll (normaler Zeilenabstand) gesetzt und es wird auf das nächste Überweisungsformular vorgeschoben. Nun müssen noch eventuell eingeschaltete Schriftarten ausgeschaltet werden.

protokoll ():

In der Prozedure ‚protokoll ()' werden die wichtigsten Daten der Überwei-

sung gespeichert. Dies geschieht, indem man einfach eine Datei zum Anhängen öffnet; ist keine vorhanden, wird sie automatisch kreiert. Damit erhält man eine Liste der gedruckten Überweisungen auf Diskette. Diese können Sie als Übersicht mit einem Textverarbeitungsprogramm weiterverarbeiten oder in eine Datenbank aufnehmen. Wem dies unnötig erscheint, kann die Prozedure ‚protokoll ()' und den Aufruf in ‚drucken ()' weglassen.

print ():

Außerdem habe ich noch eine Prozedure schreiben müssen, damit die Druckeranpassung berücksichtigt wird. Zudem kann mit dem Befehl ‚Cprnout ()' nur ein Zeichen gedruckt werden. ‚print ()' ist eine Prozedur, die es ermöglicht, einen String zu drucken und dabei nötige Konvertierungen vorzunehmen. Wenn Sie Probleme mit falschgedruckten Zeichen haben, können Sie diese in der Switch-Anweisung unterbringen. Die Variable ‚flag' dient zur Unterscheidung, welcher Zeichensatz zum Drucken eines Zeichens angewählt werden soll. Wenn flag=0 ist, muß auf den amerikanischen Zeichensatz umgeschaltet werden, und das entsprechende Zeichen kann dann gedruckt werden. Anschließend wird wieder der deutsche Zeichensatz angewählt.

akt datum ():

In dieser Prozedur wird das aktuelle Computerdatum ausgelesen und in das Überweisungsformular geschrieben. Dies ist natürlich nur sinnvoll, wenn Sie eine Echtzeituhr haben oder die im Tastaturprozessor befindliche Uhr gepuffert haben, da sonst eventuell ein veraltetes Datum erscheint.

file box ():

Anschließend möchte ich noch die Funktion ‚e=file_box (a,b,c,d) näher erläutern, da sie als nützliche Unterroutine auch in anderen Programmen verwendet werden kann.

In ‚a' übergibt man eventuell Laufwerk und/oder Pfadvorgaben („", „E:", „C:/MEGAMAX", „/1ST_WORD").

In ‚b' gibt man ein eventuelles Auswahlkriterium an („", „/*.*", „/FORMULAR.*", „/*.DAT").

Mit dem Parameter ‚c' können Sie

```
210:          case 'Ü':  string[i]=93;   flag=1;  break;
211:          case 'Ö':  string[i]=92;   flag=1;  break;
212:          case 'ä':  string[i]=123;  flag=1;  break;
213:          case 'ö':  string[i]=124;  flag=1;  break;
214:          case 'ü':  string[i]=125;  flag=1;  break;
215:          case 'ß':  string[i]=126;  flag=1;  break;
216:          case '@':  string[i]=64;   flag=0;  break; /* mit USA-Zeichensazt */
217:          case '[':  string[i]=91;   flag=0;  break;
218:          case '\\': string[i]=92;   flag=0;  break;
219:          case ']':  string[i]=93;   flag=0;  break;
220:         case '{':  string[i]=123;  flag=0;  break;
221:         case '|':  string[i]=124;  flag=0;  break;
222:         case '}':  string[i]=125;  flag=0;  break;
223:         case '~':  string[i]=126;  flag=0;  break;
224:       }
225:
226:       if (flag==0)                   /* wenn flag 0 mit USA-Zeichensatz drucken */
227:       {
228:         print("\033R\000");                     /* ESC "R" 0 (Zeichensatz USA) */
229:         Cprnout(string[i++]);                 /* ein Zeichen an Drucker senden */
230:         print("\033R\002");                /* ESC "R" 2 (Zeichensatz DEUTSCH) */
231:         flag=1;
232:       }
233:       else                               /* sonst mit DEUTSCH-Zeichensatz drucken  */
234:         Cprnout(string[i++]);                 /* ein Zeichen an Drucker senden */
235:     }
236:   }
237:
238:   /****************************************************************************/
239:
240:   protokoll()     /* Speichert Daten der Überweisung in Liste auf Diskette ab */
241:   {
242:     int i;
243:
244:     file_handle=fopen(protoname,"a");               /* Datei zum Anhängen öffnen */
245:     fprintf(file_handle,"%13s | %s\n",**p_form[12],**p_form[ 0]);
246:     fprintf(file_handle,"%13s | %s\n",""           ,**p_form[ 1]);
247:     fprintf(file_handle,"%13s | %s\n",**p_form[ 3],**p_form[ 4]);
248:     fprintf(file_handle,"%13s | %s\n",""           ,**p_form[ 5]);
249:     fprintf(file_handle,"%13s | %s\n",**p_form[ 8],**p_form[ 6]);
250:     fprintf(file_handle,"%13s | %s\n",""           ,**p_form[ 7]);
251:     fprintf(file_handle,"%13s | %s\n",**p_form[ 9],**p_form[10]);
252:     fprintf(file_handle,"%13s | %s\n",""           ,**p_form[11]);
253:     fprintf(file_handle,
254:            "-----------------------------------------------------------------\n");
255:     fclose(file_handle);                                    /* Datei schließen */
256:   }
257:
258:   /****************************************************************************/
259:
260:   akt_datum()                      /* aktuelles Datum aus der Compteruhr auslesen */
261:   {
262:     int date;
263:     char zr,datestr[11];
264:
265:     date=Tgetdate();                                          /* Datum auslesen */
266:     zr=((char)date)&0x1f;
267:     datestr[0]=zr/10+'0';                            /*  Zehnerstelle des Tages */
268:     datestr[1]=zr%10+'0';                            /*  Einerstelle  des Tages */
269:     zr=((char)(date>>=5))&0xf;
270:     datestr[2]='.';
271:     datestr[3]=zr/10+'0';                            /* Zehnerstelle des Monats */
272:     datestr[4]=zr%10+'0';                            /* Einerstelle  des Monats */
273:     zr=((char)((date>>4)+80L)%100)&0x7f;
274:     datestr[5]='.'; datestr[6]='1'; datestr[7]='9';
275:     datestr[8]=zr/10+'0';                            /* Zehnerstelle des Jahres */
276:     datestr[9]=zr%10+'0';                            /* Einerstelle  des Jahres */
277:     datestr[10]='\0';
278:     sprintf(**p_form[12],"%s",datestr);         /* Datum in Formular eintragen */
279:   }
280:
281:   /****************************************************************************/
282:
283:   file_box(path,choise,word,path_name)          /* eigene Fileselectbox-routine */
284:   char *path,*choise,*word,*path_name;
285:   {
```

```
286:     int key,ii,kk;
287:     char kette[64];
288:
289:     ii=strlen(path);                               /* Länge des Vorgabepfades */
290:     while((ii!=0)&&(path[ii]!=':'))        /* Laufwerkangaben suchen (vor ':') */
291:       ii--;
292:     if (ii==0)             /* Wenn kein ':' vorhanden dann Laufwerkangabe holen */
293:     {
294:       kette[0]=Dgetdrv()+'A';                    /* aktuelles Laufwerk auslesen */
295:       strcpy(&kette[1],":");
296:       strcat(kette,path);                      /* Laufwerk + restl. Pfadvorgabe */
297:       strcpy(path,kette);                          /* zurück auf path kopieren */
298:     }
299:     if (strlen(choise)==0)                  /* Wenn noch kein Auswahlkriterium */
300:       sprintf(choise,"\\*.*");                     /* absolutes Suchkriterium */
301:     strcat(path,choise);                               /* Kriterium anhängen */
302:
303:     fsel_input(path,word,&key);       /* eigentlicher Fileselect aufruf für GEM */
304:
305:     ii=strlen(path);                              /* nach '\' von hinten suchen */
306:     while ((ii!=0)&&(path[ii]!='\\'))
307:       ii--;
308:     path[ii]='\0';                                 /* Suchkriterium abschneiden */
309:
310:     if (key!=0)                                   /* wenn mit OK abgeschlossen */
311:     {
312:       ii=strlen(word);                            /* nach '.' von hinten suchen */
313:       while((ii!=0)&&(word[ii]!='.'))
314:         ii--;
315:       if (ii==0)                                 /* Wenn kein Extention dran ist */
316:       {
317:         ii=strlen(choise);               /* Position des '.' im Extention suchen */
318:         while(choise[ii]!='.')
319:           ii--;
320:         for (kk=ii; kk<=strlen(choise); kk++)   /* ab dieser Position bis Ende */
321:           kette[kk-ii]=choise[kk];                /* Extention kopieren und ... */
322:         kette[++kk]='\0';
323:         strcat(word,kette);                      /* ... anhängen (z.B. '.FRM')  */
324:       }
325:               /* Filename besteht aus Laufwerk + Pfad und ausgewähltem Namen */
326:       sprintf(path_name,"%s\\%s",path,word);
327:     }
328:     else
329:       path_name[0]='\0';             /* mit Abbruch beendet => kein Name gewählt */
330:     return(key);
331:   }
332:
333:   /*****************************************************************************/
334:
335:   main()           /* Hauptprogramm übernimmt Kontrolle des Programmablaufs */
336:   {
337:     int k;
338:     char drive[2];                          /* für aktuelles Laufwerk */
339:
340:     gem_init();                  /* Arbeitsstation anmelden und RSC-File laden */
341:
342:     Dgetpath(pfad,0);  /* aktuellen Pfad auslesen und als 1. Vorgabe verwenden */
343:     sprintf(wahl,"\\*.FRM");     /* Auswahlkriterium nach allen FoRMat-Dateien */
344:
345:     drive[0]=Dgetdrv()+'A';                         /* aktuelles Laufwerk holen */
346:     strcpy(&drive[1],":");
347:     sprintf(protoname,"%s%s\\PROTOKOLL.DAT",drive,pfad);       /* Protokollpfad */
348:
349:     hidemaus();
350:     pfeilmaus();
351:     xy[0]=0; xy[1]=0; xy[2]=639; xy[3]=18;
352:     vsf_interior(handle,0);
353:     vsl_color(handle,0);
354:     v_bar(handle,xy);          /* weiße Leiste an oberen Bildschirmrand zeichen */
355:     vsl_color(handle,1);
356:     v_gtext(handle,16,14,                               /* und mit Textbeschreiben */
357:   "Druckprogramm für Überweisungsformulare - programmiert von UHS-Soft März '87");
358:
359:     tree=form_addr;                                /* Pointer auf Editfelder setzen */
360:     p_form[00]=(char ***)(OB_SPEC(EMPF_1));
361:     p_form[01]=(char ***)(OB_SPEC(EMPF_2));
```

schon einen Filenamen angeben, der dann erscheint.

Auf ‚d‘ erhalten Sie dann den Filenamen mit Laufwerksangabe, Pfadangaben und dem ausgewählten Filenamen.
Schließlich gibt der Funktionswert an, ob mit OK (e=1) oder mit Abbruch (e=0) die Fileselectbox verlassen wurde.

An dieser Stelle möchte ich noch auf einen Fehler in der Betriebssystem-Routine ‚fsel_input()‘ hinweisen. Während des Aufrufs dieser Fileselect-Routine passiert es, daß bei Eingabe von ‚–‘ in der Pfadzeile das System sich mit mehreren Bomben verabschiedet. Schade, daß es sich hierbei um einen Systembug handelt, den wir nur mit einem Austausch der ROM's beseitigen können. Wenn wir den Fehler ganz umgehen wollen, müßten wir eine komplett neue Routine schreiben. Das dies aber recht aufwendig ist, habe ich mich entschieden, die alte Systemroutine weiterzuverwenden, der Fehler tritt bei normalem Gebrauch nur selten auf, und unter Berücksichtigung des Bug's dürfte es zu keinem Absturz führen.

Beschaffung des Überweisungspapiers und Einspannen in den Drucker:

Auf Ihrer Bank bekommen Sie bestimmt kostenlos die benötigten Endlos-Überweisungsformulare. Die meisten Formulare entsprechen den genormten Formularen. Beim Einspannen kann ich Ihnen den Tip geben: das Formular so einspannen, daß das zu druckende Formular (in diesem Fall das zweite) genau vor dem Druckkopf plaziert werden kann. Die genaue Position in Ihrem Drucker müssen Sie selber durch Probedrucke in Erfahrung bringen.

Und nun viel Spaß mit dem Programm und beim Drucken Ihrer Überweisungen.

Ulrich Stumm

```
362:        p_form[02]=(char ***)(OB_SPEC(BLZ));
363:        p_form[03]=(char ***)(OB_SPEC(KTO_EMPF));
364:        p_form[04]=(char ***)(OB_SPEC(BANK));
365:        p_form[05]=(char ***)(OB_SPEC(ZWECK_1));
366:        p_form[06]=(char ***)(OB_SPEC(ZWECK_2));
367:        p_form[07]=(char ***)(OB_SPEC(ZWECK_3));
368:        p_form[08]=(char ***)(OB_SPEC(BETRAG));
369:        p_form[09]=(char ***)(OB_SPEC(KTO_AUFT));
370:        p_form[10]=(char ***)(OB_SPEC(ANSCHR_1));
371:        p_form[11]=(char ***)(OB_SPEC(ANSCHR_2));
372:        p_form[12]=(char ***)(OB_SPEC(DATUM));
373:        for (k=0; k<13; k++)                          /* Editfelder mit Leerstring */
374:          sprintf(**p_form[k],"\0");                  /* vorbesetzen              */
375:
376:        form_center(form_addr,&xd,&yd,&wd,&hd);          /* Dialogbox zentrieren  */
377:        form_dial(FMD_START,xd,yd,wd,hd);                 /* Dialogbox vorbereiten */
378:        form_dial(FMD_GROW,1,1,1,1,xd,yd,wd,hd);          /* wachsendes Rechteck   */
379:        objc_draw(form_addr,0,tiefe,0,19,639,399);        /* Dialogbox zeichen     */
380:        showmaus();
381:
382:        do
383:        {
384:          knopf=form_do(form_addr,EMPF_1);                 /* GEM verwaltet Dialog  */
385:          switch(knopf)                                   /* mit welchem Knopf     */
386:          {                                               /* wurde Box beendet ??? */
387:            case LOAD:
388:                 laden();
389:                 objc_change(form_addr,LOAD,0,xd,yd,wd,hd,SHADOWED,1);
390:                 if (taste==1)                         /* wenn neue Texte vorhanden, */
391:                   objc_draw(form_addr,5,tiefe,0,19,639,399);/* dann auch ausgeben */
392:                 break;
393:
394:            case SAVE:
395:                 speichern();
396:                 objc_change(form_addr,SAVE,0,xd,yd,wd,hd,SHADOWED,1);
397:                 break;
398:
399:            case PRINT:
400:                 drucken();
401:                 objc_change(form_addr,PRINT,0,xd,yd,wd,hd,SHADOWED,1);
402:                 break;
403:
404:            case QUIT:
405:                text="[3][Wollen Sie das Programm|wirklich beenden?][beenden|zurück]";
406:                 taste=form_alert(1,text);
407:                 if (taste==1)
408:                 {
409:                   form_dial(FMD_FINISH,1,1,1,1,xd,yd,wd,hd); /* Dialogbox beenden */
410:                   form_dial(FMD_SHRINK,1,1,1,1,xd,yd,wd,hd); /* schrumpfendes Rec */
411:                   v_clsvwk(handle);                      /* Arbeitsstation schließen */
412:                   gemdos(0x0);                              /* zurüch zum Desktop */
413:                 }
414:                 objc_change(form_addr,QUIT,0,xd,yd,wd,hd,SHADOWED,1);
415:                 break;
416:
417:            case DATE:
418:                 akt_datum();
419:                 objc_change(form_addr,DATE,0,xd,yd,wd,hd,NORMAL,1);
420:                 objc_draw(form_addr,DATUM,tiefe,0,19,639,399);
421:                 break;
422:
423:          }                                                  /* end of switch */
424:        } while(1);                                          /* Endlosschleife */
425:     }                                                       /* end of main    */
```

Änderungen am Betriebssystem

Der folgende Bericht zeigt Ihnen anhand von Beispielen, wie Sie eigene Routinen in den Betriebssystem-ROMs (TOS) unterbringen können. Ferner werden Tips gegeben, um Änderungen, wie z.B. die Umstellung von 50 Hz auf 60 Hz direkt in den ROMs bzw. EPROMs vornehmen zu können.

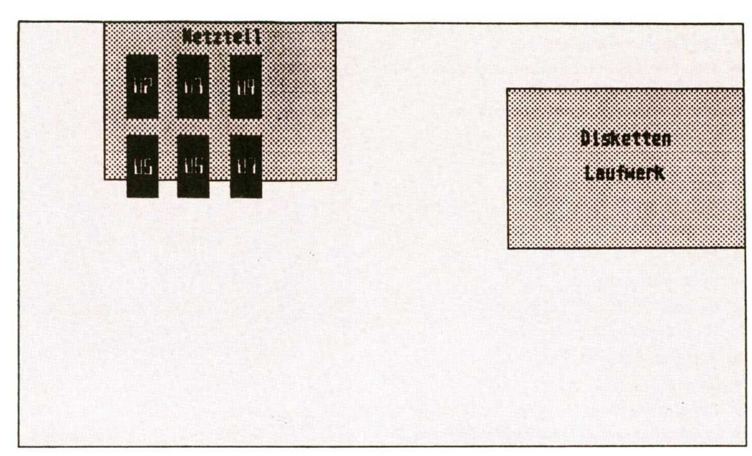


Abb. 1a: Lage der Betriebssystem-ROM's im 1040 ST/F

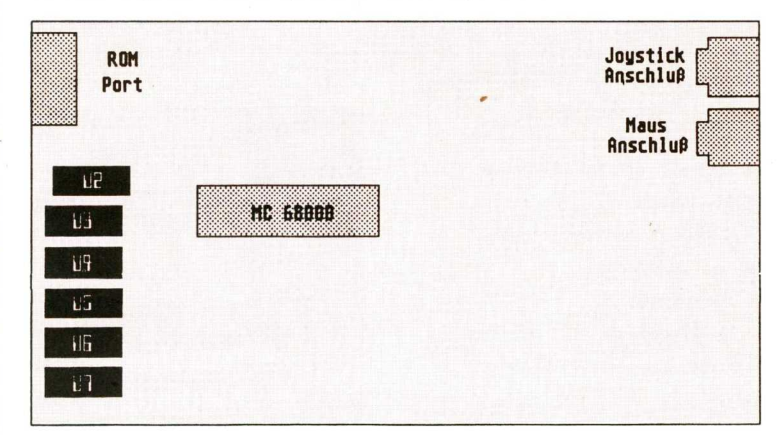


Abb. 1b: Lage der Betriebssystem-ROM's im 260 ST und 520 ST

Viele von Ihnen werden sich schon geärgert haben, daß die ST-Uhr beim Einschalten des Rechners nur dann gestellt wird, wenn sich ein entsprechendes Programm im AUTO-Ordner oder als Accessory auf der Diskette befindet. Das kann man ändern, wenn man das Boot-Programm in die Reset-Routine einbindet und dann mit dem Betriebssystem in entsprechende EPROM's brennt. Das setzt natürlich voraus, daß man in der Lage ist, EPROM's des Typs 27256 zu brennen, aber es sind schon mehrere preisgünstige Programmiergeräte erhältlich, so daß dieser Punkt für Interessierte kein Problem darstellen wird. Einige von Ihnen werden sicherlich auch folgendes Problem kennen: Sie haben einen Star, Epson oder IBM kompatiblen Drucker an Ihren Rechner angeschlossen. Wenn Sie jetzt über Alternate Help eine Hardcopy machen wollen, so geht dies nur wenn Sie vorher das Control-Accessory mitgebootet haben und dort von 1280 auf 960 Punkte/Inch umgeschaltet haben. Diese Einstellung wird zwar mit dem DESKTOP.INF auf Diskette abgespeichert, sie wird jedoch nur vom Kontrollfeld-Accessory gelesen. Ähnlich ist es mit der seriellen Schnittstelle (RS 232), auch diese Parameter werden nur von dem entsprechenden Accessory aus dem DESKTOP.INF gelesen und gesetzt.

Außerdem möchte ich Ihnen noch zeigen, wie man, auch ohne ein DESKTOP.INF auf der Diskette zu haben, entsprechende Änderungen des Bildschirmaufbaus in den TOS-ROM's erreichen kann.

Die Atari ST-Serie besitzt 6 Betriebssystem-ROM's, die mit U2-U7 bezeichnet sind. Die Lage der ROM's im 1040 ST zeigt Bild 1a. Um an die ROM's heranzukommen, muß man jedoch vorher das Netzteil abschrauben, da diese unter besagtem Netzteil liegen. Im 260 ST und im 520 ST, egal ob mit oder ohne Modulator, liegen die ROM's wie im Bild 1b dargestellt. Bei allen 3 Rechnern muß natürlich vorher das Abschirmblech entfernt werden, sonst geht garnichts.

Wie Ihnen sicher bekannt sein dürfte, gehört die Atari ST-Serie zu den 16 Bit Rechnern. Die ROM's sind aber nur 8 Bit-weise organisiert, so daß jeweils 2 ROM's ein komplettes Wort (16 Bit) darstellen. Die Verteilung der Bytes zeigt Ihnen Bild 2 am Beispiel der ROM's U4 und U7. Für die anderen ROM's gilt das analog. Das Betriebssystem befindet sich im Speicherbereich von Adresse $FC0000 bis Adresse $FEFFFF. Die Zuordnung der einzelnen ROM's zum Speicher ist wie folgt: (siehe Bsp. 1)

Im weiteren werde ich nur noch die 16 Bit Schreibweise wählen, also stets höherwertiges und niederwertiges Byte hintereinander, genauso wie die Änderungen später im Speicher stehen, und nicht nach ROM's bzw. geraden/ungeraden Adressen unterteilt. Das ist durchaus sinnvoll, da z.B. ASCII-Texte sonst keinen Sinn er-

```
Rechner Adresse FC0000 : 60 1E 01 00 00 FC 00 20 ...

  ROM U4                                    ROM U7
Adresse 0000 : 60 01 00 00 ...      Adresse 0000 : 1E 00 FC 20 ...

  ROM U4 : gerade Adresse , höherwertiges Byte
  ROM U7 : ungerade Adresse , niederwertiges Byte
```

Abb. 2: Aufteilung der 16-Bit-Adressen auf die 8-Bit-ROM's

U2	=	gerade Adresse im Bereich $FE0000 – $FEFFFF
U3	=	gerade Adresse im Bereich $FD0000 – $FDFFFF
U4	=	gerade Adresse im Bereich $FC0000 – $FCFFFF
U5	=	ungerade Adresse im Bereich $FE0000 – $FEFFFF
U6	=	ungerade Adresse im Bereich $FD0000 – $FDFFFF
U7	=	ungerade Adresse im Bereich $FC0000 – $FCFFFF

Beispiel 1

```
ALT :
FC03C4 3F3900FC001E     MOVE.W $FC001E,-(A7)  ; Datum (6.2.86) holen
FC03CA 3F3C002B         MOVE.W #$2B,-(A7)     ; Set Date
FC03CE 4E41             TRAP #1               ; GEMDOS
FC03D0 584F             ADDQ.W #4,A7 ; Stackpointer korrigieren
FC03D2 610000B8         BSR $FC048C  ; von Floppy booten
FC03D6 610000D0         BSR $FC04A8  ; von DMA-Bus booten
FC03DA 61000944         BSR $FC0D20  ; reset-residente Programme ausführen
FC03DE    hier geht die Original-Reset-Routine weiter

NEU :
FC03C4 610000C6         BSR $FC048C  ; von Floppy booten
FC03C8 610000DE         BSR $FC04A8  ; von DMA-Bus booten
FC03CC 61000952         BSR $FC0D20  ; reset-residente Programme ausführen
FC03D0 4EB900FEFC80     JSR $FEFC80  ; Sprung zum Unterprogramm
FC03D6 4E71             NOP          ; No Operation - zum Auffüllen
FC03D8 4E71             NOP          ; des Speichers bis die Original
FC03DA 4E71             NOP          ; Reset Routine weitergeht
FC03DC 4E71             NOP
FC03DE    hier geht die Original-Reset-Routine weiter
```

Listing 1

```
Zielcode

3F3C0017     MOVE.W #23,-(SP)       ; Gettime XBIOS
4E4E         TRAP #14               ; aufrufen
548F         ADDQ.L #2,SP           ; Stack reparieren
3F00         MOVE.W D0,-(SP)        ; LOW Byte = Zeit auf Stack schreiben
4840         SWAP D0                ; LOW Byte mit HIGH Byte tauschen
3F00         MOVE.W D0,-(SP)        ; HIGH Byte = Datum auf Stack schreiben
3F3C002B     MOVE.W #$2B,-(SP)      ; SET DATE GEMDOS
4E41         TRAP #1                ; aufrufen
588F         ADDQ.L #4,SP           ; Stack reparieren
3F3C002D     MOVE.W #$2D,-(SP)      ; SET TIME GEMDOS
4E41         TRAP #1                ; aufrufen
588F         ADDQ.L #4,SP           ; Stack reparieren
```

Listing 2

Bitnummer	0	1
0	Matrixdrucker	Typenraddrucker
1	Monochromdrucker	Farbdrucker
2	Ataridrucker	Epsondrucker (960 DPI)
3	Test-Modus	Qualitäts-Modus
4	Centronics-Port	RS232-Port
5	Endlospapier	Einzelblatt
6, 7	reserviert	

Beispiel 2

geben. Außerdem kennen nahezu alle gängigen EPROM-Programmiergeräte, wie z.B. der Easyprommer, eine 16 Bit Verarbeitung. Für alle, die das nicht können, gilt die Aufspaltung nach Bild 2.

Jetzt werden Sie sich sicher fragen: Wie kann man noch eigene Routinen im Betriebssystem unterbringen? Ganz einfach: Am Ende des Betriebssystems befinden sich noch einige hundert Bytes Platz, die für eigene Routinen genutzt werden können. Man muß lediglich an geeigneter Stelle einen Unterprogrammaufruf aus der Reset-Routine vornehmen. Diesen Aufruf kann man am einfachsten dort einsetzen, wo das TOS-Datum (6.2.86) gesetzt wird. Diese Routine wird ohnehim nicht mehr benötigt, wenn Sie eine ST-Uhr haben und diese in Ihrem Unterprogramm setzen wollen. Wollen Sie jedoch nur die Parameter für den Drucker oder die RS232 ändern und haben keine ST-Uhr, dann sollten Sie das TOS-Datum in Ihrem Unterprogramm setzen, sonst steht es auf 00:00:80. Leider kommen nach der Setzroutine für das Datum noch 3 Routinen, die Initialisierungsaufgaben übernehmen. Deshalb müssen diese Routinen vorverlegt werden. Anhand der folgenden Gegenüberstellung von alt und neu (Listing 1) sehen Sie, was ich meine:

Wenn Sie die entsprechenden Bytes ändern, so wird, anstatt das TOS-Datum zu setzen, zu einem Unterprogramm nach $FEFC80 verzweigt. Dort müssen nun unsere Ergänzungen und Änderungen hin. Im Original stehen von $FEFC78 bis $FEFFF3 nur Nullen, daher entfällt hier die Angabe des jeweiligen ‚alten' Inhalts. Ich habe hier auch auf die Angabe von Adressen verzichtet, da es egal ist, welche der folgenden Routinen ab der Adresse $FEFC80 stehen. Wichtig ist nur, daß die erste Routine bei $FEFC80 beginnt und zum Schluß aller Routinen ein Return to Sender (RTS) steht.

Die erste Routine liest die ST-Uhr aus der Tastatur und schreibt die Zeit und das Datum in die interne Uhr des ST (Listing 2).

Die zweite Routine setzt die Druckerkonfiguration gemäß Ihren Wünschen (Listing 3). Die Druckerkonfiguration ist ein Bitvektor mit folgender Bedeutung: (siehe Bsp. 2)

Wenn Sie also den zum Anfang genannten Fehler bei der Hardcopy-Funktion beheben wollen, so müssen sie nur Bit 2 auf 1 setzen, d.h. der Vektor hat den Wert %00000100 = $04. Wollen Sie Ihren Drucker zusätzlich an der RS232 betreiben, so muß außerdem Bit 4 auf 1 gesetzt sein, und Sie erhalten den Wert %00010100 = $14. Diesen Wert könnte man nun über die XBIOS Funktion 33 dem Rechner übergeben, es ist jedoch kürzer, den Wert gleich an die entsprechende Stelle im RAM zu schreiben.

```
Zielcode

31FC00**0E4A  MOVE.W #VEKTOR,$0E4A   ; Vektor ins RAM schreiben

Anstelle der Sternchen ** muß der Hexwert des Bitvektors stehen
z.B. 04 oder 14 gemäß obigen Beispielen
```

Listing 3

Die dritte Routine ermöglicht Ihnen die Baudrate und den Handshake-Modus der RS232 zu setzen (Listing 4). Hierbei müssen zwei Parameter beachtet werden: (siehe Bsp. 3)

```
Zielcode

3F3CFFFF        MOVE.W #-1,-(SP)     ; Diese Werte bleiben
3F3CFFFF        MOVE.W #-1,-(SP)     ; auf -1 , sonst werden
3F3CFFFF        MOVE.W #-1,-(SP)     ; die MFP-Register
3F3CFFFF        MOVE.W #-1,-(SP)     ; verändert .
3F3C00**        MOVE.W #shake,-(SP)  ; ** = Handshake Parameter shake
3F3C00**        MOVE.W #baud,-(SP)   ; ** = Baudraten Parameter baud
3F3C000F        MOVE.W #15,-(SP)     ; XBIOS-Routine
4E4E            TRAP #14             ; aufrufen
DFFC0000000E    ADD.L #14,SP         ; Stack reparieren
```

Listing 4

shake	Bedeutung
$00	kein Handshake (Default)
$01	XON/XOFF
$02	RTS/CTS
$03	XON/XOFF und RTS/CTS

und:

baud	Baudrate	*baud*	Baudrate	*baud*	Baudrate
$00	19200	$01	9600	$02	4800
$03	3600	$04	2400	$05	2000
$06	1800	$07	1200	$08	600
$09	300	$0A	200	$0B	150
$0C	134	$0D	110	$0E	75
$0F	50				

Beispiel 3

Folgendes Beispiel stellt Handshake auf XON/XOFF und Baudrate auf 300 Baud (Listing 5):

```
Zielcode

3F3CFFFF        MOVE.W #-1,-(SP)
3F3CFFFF        MOVE.W #-1,-(SP)
3F3CFFFF        MOVE.W #-1,-(SP)
3F3CFFFF        MOVE.W #-1,-(SP)
3F3C0001        MOVE.W #1,-(SP)       ; XON/XOFF
3F3C0009        MOVE.W #9,-(SP)       ; 300 Baud
3F3C000F        MOVE.W #15,-(SP)
4E4E            TRAP #14
DFFC0000000E    ADD.L #14,SP
```

Listing 5

Wenn Sie wollen, können Sie alle drei Routinen hintereinander hängen. Die Reihenfolge ist egal. Sie sollten jedoch das rechnerinterne DESKTOP.INF mit ändern, wenn Sie die RS232-Parameter oder die Druckerparameter durch eine der obigen Routinen geändert haben, da sonst das Control- bzw. RS232-Accessory diese Änderungen zunichte macht und wieder durch die Werte des rechnerinternen DESKTOP.INF ersetzt. Doch hierzu später mehr.

Wichtig:
Der letzte Befehl muß RTS sein!

Zum Abschluß Ihrer letzten Routine muß der Code 4E75 stehen, sonst kommt Ihr Rechner nicht mehr in die Reset-Routine zurück.

```
Zielcode
4E75            RTS
```

Natürlich gibt es noch vieles, was man ändern könnte: z.B. die Farben, die ebenso im DESKTOP.INF definiert werden können, aber nur vom Kontrollfeld gelesen werden. Leider setzen diese Unterprogramme Kenntnisse in Maschinensprache voraus und können nicht so einfach als Hexdump angegeben werden. Wer sich in Maschinensprache oder Assembler auskennt, kann dann natürlich auch kompliziertere Unterprogramme mit einbinden.

Tips für Änderungen am TOS

Nun möchte ich noch einige Tips zum Bildschirmaufbau geben. Dazu gehören: Änderung des rechnerinternen DESKTOP.INF, Änderung der Icons (Disk-Station, Papierkorb, Ordner...) und Beseitigen des Flimmerns auf manchen Farbmonitoren (Umstellung von 50 Hz auf 60 Hz). Für die nun folgenden Änderungen brauchen Sie **nicht** die Reset-Routine zu verändern, Sie können das zuvor gesagte also getrost vergessen.

Das DESKTOP.INF

Beginnen wir mit dem rechnerinternen DESKTOP.INF – es steht in den ROM's U3 und U6 im Speicherbereich

$FD87BC – $FD89D4 incl., das sind genau 537 Bytes. Alle Änderungen, die Sie machen, dürfen die Länge von 537 Bytes nicht verändern d.h. Sie können nicht beliebig an das Vorhandene anhängen, sondern müssen im Original nicht benötigte Definitionen entfernen und statt dessen Ihre eigenen Definitionen einfügen. Das Original sieht dabei so aus (Abbildung 3):

```
#a000000
#b000000
#c7770007000600070055200505552220770557075055507703111103
#d
#E 1B 00
#W 00 00 04 03 43 10 09 @
#W 00 00 0D 08 2A 0B 00 @
#W 00 00 0E 09 2A 0B 00 @
#W 00 00 0F 0A 2A 0B 00 @
#M 00 00 00 FF A DISKSTATION@ @
#M 00 01 00 FF B DISKSTATION@ @
#M 00 02 00 FF C HARD DISK@ @
#C 01 06 00 FF c ROM-Modul@ @
#T 00 06 02 FF   PAPIERKORB@ @
#F FF 04   @ *.*@
#D FF 01   @ *.*@
#G 03 FF   *.PRG@ @
#G 03 FF   *.APP@ @
#F 03 04   *.TOS@ @
#P 03 04   *.TTP@ @
```

Abb. 3: Das original DESKTOP.INF

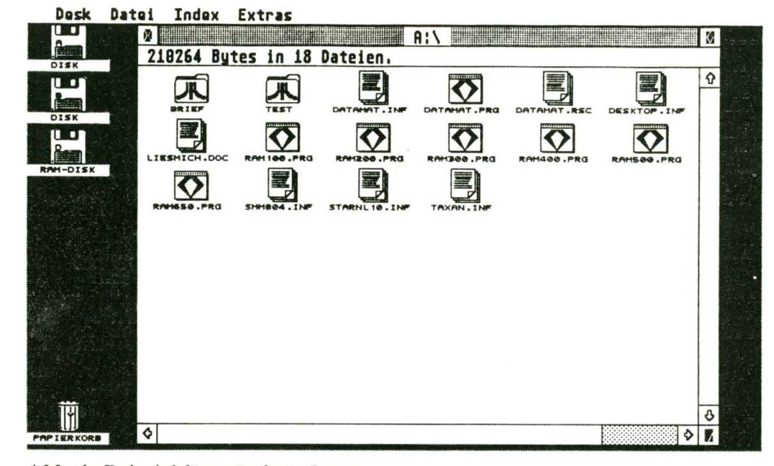


Abb. 4: Beispiel für geänderte Icons

Über die Bedeutung der einzelnen Zeilen möchte ich mich hier nicht auslassen, bitte lesen Sie dazu in der ST Computer Nr. 10/86 auf den Seiten 39 bis 41 nach. Sicher werden Sie sich jetzt fragen: Wenn im Rechner doch eine Hard Disk und ein ROM-Modul definiert sind, warum werden diese Icons nicht dargestellt? Das liegt daran, daß der Rechner beim Einschalten prüft, ob eine Hard Disk bzw. ein ROM-Modul vorhanden ist, und nur dann das entsprechende Icon darstellt. Das ist zumindest für die Hard Disk richtig. Das Icon ROM-Modul hat sich bei mir noch nie gemeldet, egal ob im ROM-Port etwas steckte oder nicht. Wenn Sie keine Hard Disk und kein ROM-Modul haben, dann können Sie diese beiden Zeilen schon für eigene Definitionen verwenden. Wenn Ihnen nur ein paar Bytes fehlen, so können Sie auch die Namen ‚DISKSTATION' und ‚PAPIERKORB' kürzen, z.B. in ‚DISK' und ‚MÜLL' das bringt auch Platz. Als weiteres können Sie die Zeile ‚#G03FF *.APP@@' entfernen, da Files mit der Endung .APP nur sehr selten vorkommen und dann meist aus anderen Programmen aufgerufen werden. Im Zweifelsfall müssen Sie eben auf die entsprechende Diskette (die mit dem .APP File) ein DESKTOP.INF schreiben, welches obige Zeile wieder enthält. In 99% aller Fälle wird diese Zeile aber nicht gebraucht. Auf keinen Fall dürfen Sie die Spaces löschen, die werden meistens benötigt! Wenn Sie im DESKTOP.INF etwas ändern wollen, gehen Sie am besten wie folgt vor: Stellen Sie alle Parameter so ein, wie Sie sie gerne nach dem Einschalten hätten. Öffnen Sie die Fenster, wenn Sie diese gleich nach dem Einschalten geöffnet haben wollen. Schieben Sie die Disk- und Papierkorbicons an die Stelle, wo Sie sie am liebsten haben. Wollen Sie eine zusätzliche Diskstation z.B. für eine RAM-Disk, so melden Sie diese mit an. Ist alles nach Ihren Wünschen, so speichern Sie das ganze mit ‚ARBEIT SICHERN' im ‚EXTRAS'-Menu ab. Jetzt können Sie an der Länge des Files ‚DESKTOP.INF' auf der Diskette sehen, ob Sie Ihre Änderung im Betriebssystem unterkriegen (Länge = 537 Bytes) oder nicht (Länge <> 537 Bytes). Ist das File länger, so müssen Sie etwas weglassen oder die Namen kürzen (s.o.). Ist das File zu kurz, so können Sie z.B. die Zeilen mit den Definitionen für die Hard Disk oder das ROM-Modul noch ergänzen. (Diese beiden Zeilen fehlen meist im File ‚DESKTOP.INF' wenn keine

```
A=16614022
B=0
Do
  Inc B
  Dpoke 1017668+B*80,Dpeek(A)
  Dpoke 1017670+B*80,Dpeek(A+2)
  Add A,4
  Exit If A>=16615302
Loop
```

Listing 6 (GFA BASIC)

Hard Disk und kein ROM-Modul vorhanden sind, und deshalb wird dann das File zu kurz). Ist das File dann immer noch zu kurz, so kann man vor und hinter die Namen der Icons (z.B. HARD DISK) noch Spaces setzen, jedoch nicht mehr als 12 Zeichen zusammen. Wenn das alles geklappt hat und das File 537 Bytes lang ist, können Sie dann da DESKTOP.INF im ROM einfach durch Ihr DESKTOP.INF auf der Diskette ersetzen. Sie brauchen dann kein DESKTOP.INF mehr auf Ihren Disketten.

Ändern der Icons

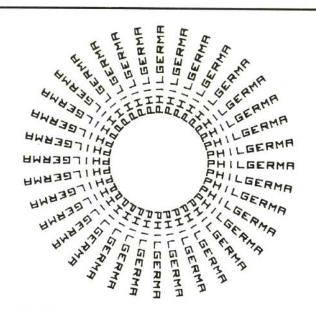
Möglicherweise haben Sie Lust, Ihre Icons zu ändern – z.B. wie in Bild 4. Die Icons bestehen aus 2 Teilen: dem Vordergrund und dem Hintergrund. Beide Teile sind je 128 Bytes lang. Die Icons stehen ebenfalls im ROM-Paar U3-U6 und zwar von $FD8286 bis $FD8785 incl. Mit einem Icon Editor kann hier der Interessierte beliebige Bildchen einsetzen. Folgendes kleine GFA-Basic Programm zeigt Ihnen die Icons (Vorder- und Hintergrund) im ROM auf dem Bildschirm (Listing 6):

50–60 Hz Umschaltung

Zum Schluß hier noch die Umstellung von 50 Hz auf 60 Hz, um Flimmern auf manchen Farbmonitoren zu unterdrücken. Dazu muß nur ein Byte im ROM U7 geändert werden. Dieses Byte steht im Speicher in der Adresse $FC0099:

Original ROM's	50 Hz	Code : $06
	60 Hz	Code : $0E

C. Hahn



PHILGERMA IHR SPEZIALIST FÜR ATARI ST COMPUTERSPRACHEN

LATTICE C (Metacomco) – Bewährter C-Compiler der USA-Firma Lattice, Standardprodukt in der IBM-Welt. Kompatibel auf vielen Rechnern, gut für professionelle Entwicklungen. Auch sehr gut für den C-Neuling da voller Kerningham/Ritchie Standard. Fließkommaarithmetik mit 16 Stellen Genauigkeit. Selbstverständlich VDI und AES Bibliothek. Ausführl. deutsches Handbuch. Sonderangebut Juni .... DM 228,00

MCC ASSEMBLER (Metacomco) – Professioneller Makro Assembler, der den vollen Motorola 68000 Instruktion Set unterstützt. Mit Editor, Linker und TOS-Libraries, AES, VDI. Ausführliche Fehlermeldungen. Handbuch .... DM 168,00

CAMBRIDGE LISP (Metacomco) – Interpreter und Compiler mit dem Sprachumfang, den man von Großrechnern gewöhnt ist. Volle Real-Arithmetik 16 MByte Adressraum. Handbuch 330 Seiten .... DM 490,00

MCC PASCAL (Metacomco) – Pascal Compiler ISO 7185 Standard. Single Pass Compiler, schnell u. effizient. Komplette Grafikeinbindung durch GEM VDI und AES Funktionen. MCC Pascal Programme können mit MCC Assembler oder Lattice C gelinkt werden. Sonderangebot Juni .... DM 98,00

BCPL Interpreter (Metacomco) – Eine leistungsfähige Sprache, besonders für die Systemsoftwareentwicklung. Mit GEM-Bibliothek .... DM 348,00

APL 68000 Interpreter (MicroAPL) – Eine sehr schnelle Version dieser von IBM-Rechnern bekannten Sprache. Info anfordern .... DM 490,00

PRO PASCAL (Prospero) – Professioneller Pascal Compiler ISO 7185/ANSI/770x3.97. Viele Erweiterungen gegenüber ISO. Fließkommaarithmetik mit 16 Stellen, dynamische Strings. Einbindung von Assembler und Fortran 77. Dieser Compiler erzeugt sehr schnellen Code auch bei großen Programmen. Handbuch 250 Seiten .... DM 448,00

PRO FORTRAN (Prospero) – ANSI X3.9 – 78 Standard Fortran 77 Compiler für 68000 Prozessoren. Optimierender zwei Pass Compiler mit vielen zusätzlichen Optionen. Fließkommaarithmetik 16 Stellen nach IEEE Standard. Große Programme durch 24 bit Adressraum. Assembler und Pro Pascal kann eingebunden werden. Ein Fortran wie auf Großrechnern. Handbuch 250 Seiten .... DM 490,00

MODULA II (TDI) – Diese umfangreiche Modula Implementierung vereinigt die Vorteile von Pascal mit neuen Sprachelementen. Integriertes Programmentwicklungssystem. Compiler, Editor, Debugger und GEM Einbindung V 3.0 neu .... DM 248,00

Sommer Sonderangebot!

Qualitätsdiskettenlaufwerke **CUMANA**
Einzellaufwerk 3'5 720 KB .... DM 498,00
Doppellaufwerk 3'5 2★720 KB ..... DM 798,00

Desktop Publishing mit:
FLEET STREET EDITOR .... DM 348,00
PUBLISHING PARTNER .... DM 498,00

Endlig FLIGHT II s/w u. Farbe ..... DM 119,00
K-SPREAD 2 Tabellenkalkulation ... DM 228,00
K-GRAPH 2 Grafik + Statistik ..... DM 148,00
K-COMM 2 Terminalprogramm ..... DM 148,00
MS-DOS Emulator Robotek .... DM 198,00

NATÜRLICH HABEN WIR AUCH SPIELE
Goldrunner DM 59,00; Psion CHESS DM 69,00; World Games DM 59,00; DeepSpace DM 89,00; Starglider DM 59,00; Alternate Reality DM 69,00 PAWN DM 69,00; Mercenary DM 49,00; Programm des Lebens DM 98,00; Skyplot DM 198,00; Major Motion DM 49,00; Karate Kid II DM 69,00
Alle Infocom Text-Adventures vorrätig

MODULA 2 Developer neu V3.0 ...... DM 448,00
MODULA 2 Commercial neu V3.0 .... DM 898,00
ST PASCAL plus Compiler .... DM 248,00
MEGAMAX C Compiler .... DM 448,00
GFA BASIC Interpreter .... DM 169,00
GFA BASIC Compiler .... DM 169,00
LDW BASIC Compiler .... DM 159,00
OMIKRON BASIC Interpreter .... DM 229,00
SALIX PROLOG Interpreter .... DM 198,00
FORTH ST von Data Becker .... DM 99,00
OS-9 Betriebssystem mit umfangreicher Software von Cumana. Info anfordern . DM 1098,00

ADIMENS Datenbank deutsch .... DM 468,00
dBMAN Datenbank deutsch .... DM 399,00
oder die fertigen Anwenderlösungen
dBMAN-Literatur .... DM 189,00
dBMAN-Adressen, Serienbriefe ...... DM 89,00

TEMPUS der schnelle Editor ...., DM 79,00
MONOSTAR s/w Grafikprogramm .... DM 99,00
GFA-DRAFT CAD-Programm .... DM 298,00

1st Wordplus deutsch.............. DM 199,00
BECKERtext ST Textprogramm ...... DM 199,00
SIGNUM Text- und Grafik-Programm .. DM 428,00

AUSZUG AUS UNSERER HARDWARE-LISTE
Einzellaufwerk 5 1/2 40/80 Spuren .... DM 678,00
Speichererweiterung auf 2,5 MByte ... DM 798,00
10 Disketten 3'5 2DD Klarsichtbox .... DM 59,00

HARDDISK XEBEC 40 MB, 40mS, 8 Partitions mögl.
Bausatz...DM 3.198, anschlußfertig...DM 3.698

Fordern Sie unsere umfangreiche **Preisliste** an.
Bestellungen bitte an:

PHILGERMA GmbH, Ungererstraße 42
8000 München 40, Tel: 0 89 / 39 55 51

Telefonische Bestellungen 10.00 bis 18.30 Uhr. Bestellungen unter DM 200, Versandkostenanteil DM 4,80. Nachnahme DM 3,20. Lieferung ins Ausland nur gegen Vorkasse.
Besuchen Sie unseren Softwareladen in der Ungererstraße 19. Sie können alle Produkte anschauen und testen.

PADERCOMP – Walter Ladz

Erzbergerstr. 27 · 4790 Paderborn · Tel. 0 52 51 - 3 63 96

FLOPPYSTATIONEN

PADERCOMP FL 1 .... **448,–**
3.5'', 1 Mb, eingeb. Netzteil, NEC-Laufwerk, Abm. 240x105x40 mm, anschlußfertig, graues Metallgeh. Testbericht ATARI-Magazin 2/87, Seite 70

PADERCOMP FL 2 .... **748,–**
Doppellaufwerk übereinander, sonst wie FL 1

PADERCOMP FL 3 .... **398,–**
Zweitlaufwerk für Amiga

NEC FD 1036 A 3.5'', 1 MB, 32 mm Bauhöhe ... **269,–**
dto. ST modifiziert **289,–**
Industrie Floppystecker **9,90**
ST Kabel an Shugart-Bus 3.5'' **29,90**
Monitorstecker **7,90**

ZUBEHÖR

3.5'' Disketten, ab 50 Stück .... **Superpreise!**
Disk Box SS-50, f. 50 3.5'' Disketten **19,90**
Media Box 1, f. 150 St. 3.5'' Disketten **39,90**
Druckerkabel ST **34,90**
Dataphon S21/23, 300 bzw. 1200/75 Baud, BTX **329,–**
Orion Farbmonitor CCM 1280 m. Kabel an Atari 260/520 **848,–**
NEC Multisync, alle drei Auflösungen **1798,–**
Monitor-Ständer dreh-, schwenk- und kippbar **29,–**
Joy-Star, Super Joystick mit 6 Microschaltern **22,90**
Preisliste .... **kostenlos**

DRUCKER

STAR NL 10 incl. Interface, Dt. Handbuch **748,–**
Citizen 120D 120 Z/s, NLQ **528,–**
OKIDATA ML 192 incl. vollaut. Einzelblatteinzug ... **1198,–**
OKI-Laserline 6⁺,Centronics **4495,–**
Mitsubishi DX-180W, 180 Z/s, NLQ, DIN A3 .... **998,–**
Panasonic KX-P 1091, 120 Z/s, NLQ **699,–**

Ein Schriftbild, fast wie gesetzt!
24-Nadeldrucker

NEC P6, 24 Nadeln, 216 Z/s, DIN A 4 .... **1198,–**
NEC P7, 24 Nadeln, 216 Z/s, DIN A 3 .... **1598,–**
STAR NB 24/15, 24 Nadeln, 216 Z/s, DIN A 3 ...... **1998,–**

Bestellungen per Nachnahme oder Vorkasse ab 30 DM. Auslandslieferungen nur gegen Vorkasse. Eingetragenes Warenzeichen: ATARI ST. Die Preise können günstiger liegen. Rufen Sie an! Händleranfragen erwünscht.

Dialogboxen in GFA-Basic (Teil I)

In dieser neuen Serie der ST Computer wollen wir auf die vielfältigen Fragen und Reaktionen auf unseren kleinen Artikel in der Februar-Ausgabe dieses Jahres eingehen. Damals wollten wir nur als Anregung zeigen, wie man auch in GFA-BASIC Ressourcen laden und damit Dialogboxen erstellen kann. Diese Serie soll nun tiefer in die Welt dieser kontaktfreudigen Boxen eindringen und so manchen Tip und Kniff verraten, mit dem man sein eigenes Programm verschönern kann. Dazu werden nach und nach die notwendigen und manchmal auch nur vereinfachenden Prozeduren abgedruckt, und anhand von Beispielen die Handhabung ausführlich erklärt.

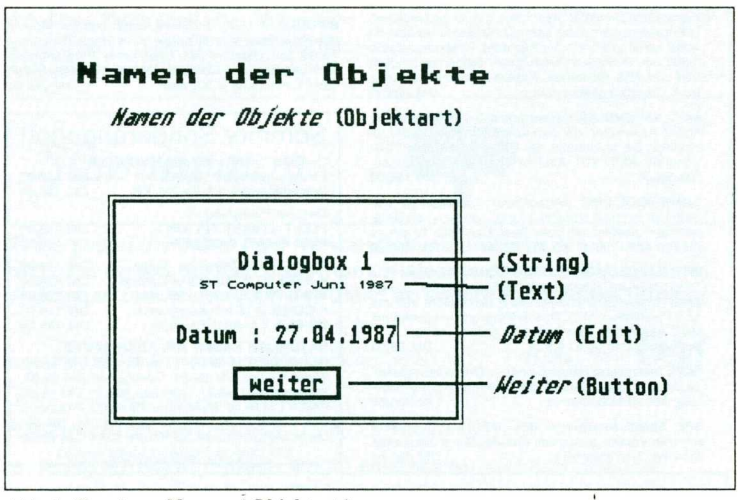


Abb. 1: Kursiv = Namen (Objektart)

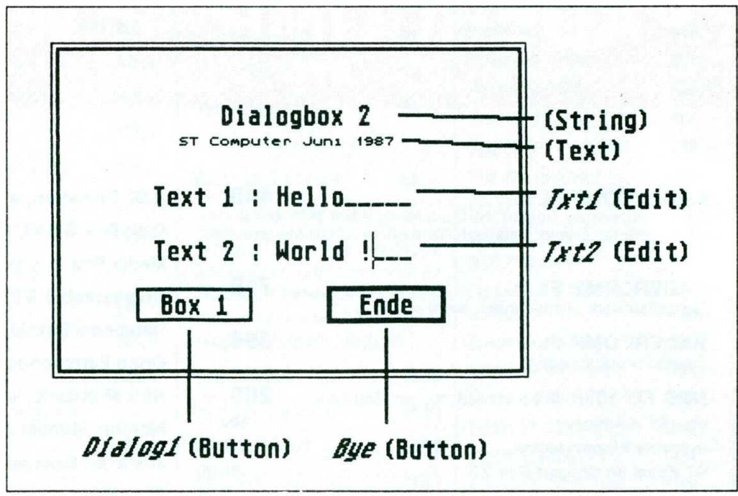


Abb. 2: Kursiv = Namen (Objektart)

Zu Beginn sollen in dieser ersten Folge die wichtigsten Routinen zur Benutzung von Dialogboxen erklärt werden. Das sind natürlich die Prozeduren, die die Boxen erst auf dem Bildschirm erscheinen und verschwinden lassen und die einzelnen Objekte abfragen. Dazu ist als Beispiel ein kleines Programm entstanden, das zwei Dialogboxen verwaltet und mehrere EDIT-Felder beinhaltet. Doch zuerst müssen die beiden Boxen in einem Resource Construction Set (RCS) erstellt werden. Wir werden nicht im einzelnen erklären, wie dies bewerkstelligt wird, da wir sonst eine Anleitung für ein RCS schreiben müssen, sondern möchten für jemanden, der noch nie mit einem RCS gearbeitet hat, auf unser nächstes Sonderheft hinweisen, in dem eine ausführliche Anleitung zum RCS V 1.4 aus dem ATARI Entwicklungspaket gegeben wird. Dieses RCS ist auch direkt bei der ST Computer gegen einen Kostenbeitrag von DM 15,— zu beziehen und somit für jedermann zugänglich.

Doch wir lassen Sie nicht ganz im Stich. Die Abbildungen 1 und 2 zeigen, aus welchen Objekttypen sich die beiden Dialogboxen zusammensetzen, so daß man sie mit ein wenig Übung auch als Anfänger leicht „nachbauen" kann.

Wichtig beim Erstellen der Resourcen ist vor allen Dingen, daß man den Objekten, die man später abfragen oder auch verändern will, einen Namen gibt. Da jedes Objekt der Resourcen einen Index bekommt, wäre dies nicht unbedingt notwendig, erleichtert die Arbeit aber ungemein, wenn man das Resource verändern muß. In diesem Falle können sich nämlich die Reihenfolge der Indizes ändern und man müßte diese für jedes Objekt im Programmlisting anpassen. Um dies zu umgehen, sind am Anfang die benötigten Namen als Variablen definiert. Wir haben dazu im RCS die Namen an die Objekte vergeben und die beim Abspeichern der

```
' ********************************
' *                              *
' *   Dialogboxen in GFA-BASIC   *
' *                              *
' *           Teil I             *
' *                     HE       *
' *                              *
' ********************************
'
Baum1=0 !      TREE
Baum2=1 !      TREE
Datum=1 !      OBJECT in TREE #0
Weiter=4 !     OBJECT in TREE #0
Txt1=1 !       OBJECT in TREE #1
Txt2=2 !       OBJECT in TREE #1
Dialog1=6 !    OBJECT in TREE #1
Bye=5 !        OBJECT in TREE #1
'
Gosub Rsrc_load("juni87.rsc",*Fehler%)
If Fehler%=0 Then
  Print "RSC-Ladefehler"
  End
Endif
'
Gtype%=0
Gindex%=Baum1
Gosub Rsrc_gaddr(Gtype%,Gindex%,*Baum1_adr%)
Obj_index%=Datum
Flag%=0          ! Cursor nicht zurücksetzen
Gosub Ptadr(Baum1_adr%,Obj_index%,Flag%,*Txt_adr%)
'
While Ex_obj%<>Bye
  Gosub Box_draw(Baum1_adr%,0,0,0,0)
  Gosub Form_do(Baum1_adr%,Datum,*Ex_obj%)
  Gosub Box_undraw(Baum1_adr%,0,0,0,0)
  Gosub Undo_objc(Baum1_adr%,Ex_obj%,1)
  Gosub Get_dialogtext(Txt_adr%,*Dat$)
  Datum$=Left$(Dat$,2)+"."+Mid$(Dat$,3,2)+"."+Right$(Dat$,4)
  Print Datum$
  '
  Gtype%=0 ,
  Gindex%=Baum2
  Gosub Rsrc_gaddr(Gtype%,Gindex%,*Baum2_adr%)
  Obj_index%=Txt1
  Flag%=1          ! Cursor zurücksetzen
  Gosub Ptadr(Baum2_adr%,Obj_index%,Flag%,*Txt_adr2%)
  Obj_index%=Txt2
  Gosub Ptadr(Baum2_adr%,Obj_index%,Flag%,*Txt_adr3%)
  '
  Gosub Box_draw(Baum2_adr%,0,0,0,0)
  Gosub Form_do(Baum2_adr%,Txt1,*Ex_obj%)
  Gosub Box_undraw(Baum2_adr%,0,0,0,0)
  Gosub Undo_objc(Baum2_adr%,Ex_obj%,1)
Wend
Gosub Get_dialogtext(Txt_adr2%,*Txt1$)
Print Txt1$
Gosub Get_dialogtext(Txt_adr3%,*Txt2$)
Print Txt2$
'
Gosub Rsrc_free
End
'
' P R O C E D U R E S
'
Procedure Rsrc_load(Rscname$,Aes_return%)
  '
  Lpoke Addrin,Varptr(Rscname$)
  Gemsys 110
  *Aes_return%=Dpeek(Gintout) !Fehlerabfrage
Return
'
Procedure Rsrc_gaddr(Gtype%,Gindex%,Aes_adr%)
  '
  Dpoke Gintin,Gtype%
  Dpoke Gintin+2,Gindex%
  Gemsys 112
  Aes_return%=Dpeek(Gintout) !Fehlerabfrage
  *Aes_adr%=Lpeek(Addrout)
Return
'
Procedure Form_center(Baum%,F_cx%,F_cy%,F_cw%,F_ch%)
  '
  Lpoke Addrin,Baum%
  Gemsys 54
  *F_cx%=Dpeek(Gintout+2)
  *F_cy%=Dpeek(Gintout+4)
  *F_cw%=Dpeek(Gintout+6)
  *F_ch%=Dpeek(Gintout+8)
Return          ,
'
Procedure Form_dial(Flag%,Littlx%,Littly%,Littlw%,Littlh%,Bigx%,Bigy%,Bigw%,Bigh%)
```

Resourcen erzeugte Header-Datei (.H) für GFA-BASIC umgewandelt (siehe ST-Ecke Mai '87) und anschließend zum Programm „hinzugemergt". Wichtig bei der Namensgebung ist, daß man keine Namen vergibt, die das BASIC als Befehl erkennt, da sonst ein Pfeil vor dem Namen erscheint, um so auf einen Fehler hinzuweisen.

Hat man nun das Resource erstellt und die Variablen am Anfang des Programmes erhalten, kann man sich dem eigentlichen Programmieren der Dialogboxen zuwenden. Dies möchten wir anhand der verwendeten Prozeduren genau erklären.

Verwendete Prozeduren und ihre Parameter

Rsrc_ — Laden einer Resource-Datei

Rsrc_("Programmname.rsc",*Fehler%)

Durch Aufruf der AES-Routine Rsrc_load (Gemsys 110) wird im Arbeitsspeicher Platz für die zu ladende Resource-Datei geschaffen und diese anschließend geladen. Als Parameter wird nur der Programmname benötigt, wobei unsere Version hier nur eine Resource-Datei in der obersten Directory-Ebene findet. Will man sie der Übersicht halber trotzdem aus einem Ordner laden, muß man zuvor mit dem CHDIR-Befehl den richtigen Pfad bestimmen.

Als Rückgabeparameter erhält man in der Variablen Fehler% eine Null oder einen positiven Wert. Ist der Wert Null, so ist ein Resource-Ladefehler aufgetreten.

Rsrc_gaddr — Ermitteln der Baumadresse

Rsrc_gaddr(0,Index%,*Baumadresse)

Nach dem Laden der Resource-Datei wird als nächstes die Baumadresse, d.h. die Stelle der Objektstruktur, an der der Baum beginnt, benötigt. Dazu verwendet man die Routine Rsrc_gaddr (Gemsys 112). Der erste Parameter (Gtype% genannt) muß in unserem Fall eine Null enthalten. Die anderen Möglichkeiten dieses Parameters sollen uns an dieser Stelle nicht interessieren. In unserem Februarheft dieses Jahres hatten wir, wie bereits oben erwähnt, einen kleinen Beitrag über Dialogboxen in GFA BASIC veröffentlicht und darin noch den Parameter Gtype% mit dem Wert 2 benutzt, um die Tedinfo-Adresse herauszubekommen. Diese Methode hat sich aber bei der Benutzung umfangreicherer Resource-Dateien als unbrauchbar erwiesen. Als universeller kann

man die jetzt verwendete Routine bezeichnen, wie sie in der Prozedur Patadr später noch beschrieben wird.

Der zweite Parameter beinhaltet die sogenannte Baumindexnummer. Will man mehrere Objektbäume in einer Resource-Datei benutzen (dies tritt ja schon bei der Benutzung zweier Dialogboxen auf), so vergibt das Resource Construction Set für jeden Baum eine Indexnummer. Man muß also die Indexnummer des gewünschten Baumes übergeben. In unserem Falle eine Null (Baum1) oder eine Eins (Baum2).

Als Rückgabeparameter erhält man die gewünschte Baumadresse zur weiteren Verwendung.

Form_center — Ermitteln der Zentrierungskoordinaten

Form_center(Baumadresse%,*F_cx%, *F_cy%,*F_cw%,*F_ch%)

Diese Routine (Gemsys 54) dient, wie der Name schon sagt, der Zentrierung einer Box in der Bildschirmmitte. Man übergibt einfach nur die Baumadresse und das AES antwortet einem freundlicherweise und meistens korrekt mit den Koordinaten für X (F_cx%), Y (F_cy%), Breite (F_cw%) und Höhe (F_ch%) der mittig zentrierten Dialogbox. Natürlich kann man die Form_center-Routine auch weglassen und eigene Koordinaten nehmen, wenn man die Dialogbox nicht in der Mitte haben will, sondern zum Beispiel in der linken unteren Ecke.

Form_dial — Öffnen und Schließen der Dialogbox

Form_dial(Flag%,Littlx%,Littly%, Littlw%,Littlh%,Bigx%,Bigy%,Bigw%, Bigh%)

Die Routine Form_dial bietet vier verschiedene Möglichkeiten des Aufrufs. Welche der vier Möglichkeiten benutzt werden soll, wird durch den ersten Parameter Flag% bestimmt. Im einzelnen bewirken sie beim Aufruf der Routine folgendes:

Flag% = 0 — reserviert einen Bildschirmbereich für die Dialogbox
Flag% = 1 — zeichnet ein sich vergrößerndes Rechteck
Flag% = 2 — zeichnet ein sich verkleinerndes Rechteck
Flag% = 3 — gibt den reservierten Bildschirmbereich wieder frei

Neben dem Flag%-Parameter verlangt Form_dial noch acht weitere Übergabewerte. Die ersten vier (Littl..) davon bil-

```
 '
  Dpoke Gintin,Flag%
  Dpoke Gintin+2,Littlx%
  Dpoke Gintin+4,Littly%
  Dpoke Gintin+6,Littlw%
  Dpoke Gintin+8,Littlh%
  Dpoke Gintin+10,Bigx%
  Dpoke Gintin+12,Bigy%
  Dpoke Gintin+14,Bigw%
  Dpoke Gintin+16,Bigh%
  Gemsys 51
  Aes_return%=Dpeek(Gintout) !Fehlerabfrage
  If Flag%=3
    Put Bigx%,Bigy%,Rette$
  Endif
Return
'
Procedure Objc_draw(Tree%,Startob%,Depth%,Xclip%,Yclip%,Wclip%,Hclip%)
  Get Xclip%,Yclip%,Xclip%+Wclip%+1,Yclip%+Hclip%+1,Rette$
  Dpoke Gintin,Startob%
  Dpoke Gintin+2,Depth%
  Dpoke Gintin+4,Xclip%
  Dpoke Gintin+6,Yclip%
  Dpoke Gintin+8,Wclip%
  Dpoke Gintin+10,Hclip%
  Lpoke Addrin,Tree%
  Gemsys 42
  Aes_return%=Dpeek(Gintout) !Fehlerabfrage
Return

Procedure Form_do(Tree%,Startob%,Aes_return%)
  '
  Dpoke Gintin,Startob%
  Lpoke Addrin,Tree%
  Gemsys 50
  *Aes_return%=Dpeek(Gintout) ! Exit_Objekt
Return
'
Procedure Ptadr(Tree%,Obj_index%,Flag%,Txt_adresse)
  Te_ptext=0
  Obj_adresse%=Tree%+24*Obj_index%
  Tedinfo_adresse%=Lpeek(Obj_adresse%+12)
  Txt_adresse%=Lpeek(Tedinfo_adresse%+Te_ptext)
  If Flag%=1 Then
    Poke Txt_adresse%,0
  Endif
  *Txt_adresse=Txt_adresse%
Return
'
Procedure Get_dialogtext(Txt_adr%,Txt%)

  I=0
  Txt$=""
  While Peek(Txt_adr%+I)<>0
    Zeichen=Peek(Txt_adr%+I)
    Txt$=Txt$+Chr$(Zeichen)
    Inc I
  Wend
  *Txt%=Txt$
Return
'
Procedure Box_draw(Tree%,X%,Y%,W%,H%)
  '
  Gosub Form_center(Tree%,*Fo_cx%,*Fo_cy%,*Fo_cw%,*Fo_ch%)
  Gosub Form_dial(0,X%,Y%,W%,H%,Fo_cx%,Fo_cy%,Fo_cw%,Fo_ch%)
  Gosub Form_dial(1,X%,Y%,W%,H%,Fo_cx%,Fo_cy%,Fo_cw%,Fo_ch%)
  Gosub Objc_draw(Tree%,0,8,Fo_cx%,Fo_cy%,Fo_cw%,Fo_ch%)
Return
'
Procedure Box_undraw(Tree%,X%,Y%,W%,H%)
  '
  Gosub Form_center(Tree%,*Fo_cx%,*Fo_cy%,*Fo_cw%,*Fo_ch%)
  Gosub Form_dial(2,X%,Y%,W%,H%,Fo_cx%,Fo_cy%,Fo_cw%,Fo_ch%)
  Gosub Form_dial(3,X%,Y%,W%,H%,Fo_cx%,Fo_cy%,Fo_cw%,Fo_ch%)
Return
'
Procedure Undo_objc(Tree%,Obj_index%,Maske%)
  '
  Obj_adresse%=Tree%+24*Obj_index%
  Aus=Dpeek(Obj_adresse%+10)
  Dpoke Obj_adresse%+10,Aus And Not Maske%
Return
'
Procedure Rsrc_free
  '
  Gemsys 111
  Aes_return%=Dpeek(Gintout) ! Fehlerabfrage
Return
```

den die X-, Y-, Breiten- und Höhenkoordinaten des Rechtecks in der Ursprungsgröße. D.h., von dieser Rechtecksgröße ausgehend wird das Rechteck auf die gewünschte Endgröße gebracht. Diese Endgröße wird wiederum von den letzten vier Parametern (Big..) bestimmt. Die Big-Parameter sollten also, will man die Dialogbox zentriert auf dem Bildschirm haben, identisch mit den Rückgabeparametern der Form_center-Routine sein.

Bildschirminhalt zerstört?

Normalerweise würde der Aufruf einer Dialogbox ein häßliches Loch in der Größe der Box auf dem Bildschirm hinterlassen, da AES den Bildschirminhalt nicht automatisch rettet. Deswegen haben wir in zwei Routinen (Form_dial und Objc_draw) zusätzlich noch ein Retten (Get) und wieder Ausgeben (Put) des überzeichneten Bildschirms eingebaut. Somit wird der Bildschirminhalt automatisch wieder regeneriert.

Es ist nicht notwendig, alle vier Form_dial-Routinen zu benutzen. Will man zum Beispiel auf den schönen Effekt des sich vergrößernden Rechtecks verzichten, so läßt man einfach die Aufrufe der Routinen mit den Parametern Flag%=1 und Flag%=2 weg. Die Dialogbox wird dann später direkt auf den reservierten Bildschirm gezeichnet. Doch dazu benötigen wir die nächste Routine.

Objc_draw — Zeichnen der Dialogbox

Objc_draw(Baumadresse%,Startobjekt%, Tiefe%,X%,Y%,W%,H%,)

Hiermit wird endlich die Dialogbox auf den Bildschirm gezeichnet. Die Routine benötigt wieder eine ganze Reihe von Parametern, die folgende Bedeutung haben. Zuerst muß natürlich, wie schon gewohnt, die Baumadresse übergeben werden. Der zweite Parameter (Startobjekt%) gibt an, ab welchem Objekt die Dialogbox gezeichnet werden soll. Wer Lust hat, kann ja mal ausprobieren, was passiert, wenn man die Box nicht ab dem ersten Objekt des Baumes (Objekt 0), sondern vielleicht erst ab dem zweiten oder dritten zeichnen läßt.

Der nächste Parameter gibt die Zeichentiefe an. Darunter ist die Anzahl der Zeichenbenen zu verstehen. AES kann maximal neun dieser Ebenen (0—8) verwalten. Alle Versuche, mehr Ebenen zeichnen zu lassen, wird prompt bombig beantwortet. Am besten gibt man als Parameter immer eine Zeichentiefe von 8 an (also neun Ebenen). So braucht man sich um diesen Parameter nie mehr Sorgen zu machen und kann immer die maximale Tiefe nutzen.

Natürlich braucht man zum Zeichnen auch die Bildschirmkoordinaten der grwünschten Dialogbox, die durch X%, Y&,W%, und H%, definiert sind. Man braucht sie aber nicht selbst auszurechnen oder auszutesten. Durch die Form_center-Routine hat man sie in der Regel schon erhalten und muß sie so nur an die Objc_draw-Routine übergeben. Selbstverständlich ist auch hier der Manipulation Tür und Tor geöffnet, so daß man leicht gewollte/ungewollte Effekte erzielen kann.

Form_do — Kontrollübergabe an AES

Form_do(Baumadresse%,Startobjekt%, *Exitobjekt%)

Durch die Form_do-Routine liefert man sich auf Gedeih und Verderb dem Rechner aus, da durch den Aufruf die komplette Kontrolle an das AES übergeben wird. Der ST wartet nun auf eine bzw. mehrere Eingabe(n) in der Dialogbox und läuft solange in einer Schleife, bis ein Objekt, das als EXIT, TOUCHEXIT oder DEFAULT definiert wurde, angeklickt wurde. Doch dazu kommen wir gleich noch nach der Parameterbeschreibung.

Als Parameter brauchen wir zuerst wieder die Baumadresse, um den richtigen Baum auszuwählen. Doch schon der nächste Parameter (Startobjekt%) benötigt eine genauere Erklärung. Er gibt an, welches Objekt der Box das erste editierbare Feld ist. Doch Vorsicht! Man sollte darauf achten, daß das Objekt auch wirklich editierbar ist, da der ST sonst gerne als Bombenleger in Erscheinung tritt. Beinhaltet die Dialogbox kein editierbares Feld (z.B. eine Info-Box), so übergibt man einfach den Wert Null.

Als Rückgabeparameter erhält man von der Form_do-Routine den Index des Objektes zurück, durch dessen Anklicken die Dialogbox verlassen (und somit die Form_do-Routine beendet) wurde. Man braucht also nur die Objektindizes mit dem Rückgabeparameter zu vergleichen und weiß sofort, durch welches Objekt die Dialogbox verlassen wurde (z.B. ABBRUCH oder OK). Wie oben schon erwähnt, muß ein solches Objekt zuvor im Resource Construction Set als EXIT, TOUCHEXIT oder DEFAULT definiert werden, da die Form_do-Routine sonst in einer Schleife läuft und ewig wartet. Man kann ein Objekt auch erst später im Programm entsprechend definieren, doch dazu kommen wir erst in der nächsten Ausgabe.

Ptadr — Ermitteln der Textadresse

Patadr(Baumadresse%,Objektindex%, Flag%,*Textadresse%)

Diese Routine ist nicht im AES enthalten und müßte somit selbst geschrieben werden. Sie erleichtert die Handhabung von EDIT-Feldern aber ungemein, da sie die Anfangsadresse eines durch den Objektindex definierten Textfeldes berechnet und zurückgibt. Benötigt werden als Parameter wieder unsere Baumadresse und der Index des gewünschten Objektes.

Die Berechnung geht folgendermaßen vonstatten: Zuerst benötigt man die eigentliche Adresse des Objektes in der Baumstruktur. Man erhält sie, indem man den Objektindex mit 24 multipliziert und das Produkt anschließend zu der Baumadresse addiert. Nun kommt sicherlich die Frage auf, wie diese Berechnung zustandekommt.

Jedes Objekt in der Baumstruktur besitzt einen 24 Byte großen Eintrag, durch den es definiert ist. Wie dieser Eintrag im einzelnen aufgebaut ist, werden wir in der nächsten Ausgabe ausführlich erklären und uns mit ihrer Benutzung beschäftigen.

Diesmal benötigen wir nur die Bytes 12, 13, 14 und 15 der Tabelle für die Ptadr-Routine, um die Adresse der sogenannten TEDINFO zu bekommen. Byte 12, 13, 14 und 15 beinhalten einen Zeiger auf die Objektspezifikation, d.h. hier steht die Adresse für die TEDINFO. Da der ATARI ST intern mit 32-Bit arbeitet, benötigen wir vier Bytes — also ein Langwort — und müssen einen LPEEK-Befehl auf die Summe von Objektadresse und 12 verwenden. Haben wir diese Adresse nun erhalten, stellt sich zunächst einmal die Frage, was ist überhaupt eine TEDINFO und wozu braucht man sie?

Box_undraw — Schließen der Dialogbox (Zusammenfassung)

Box_undraw(Baumadresse%, X%,Y%,W%,H%)

Diese Routine bildet das genaue Gegenteil zur Box_draw-Routine. In ihr werden alle nötigen Aufrufe zum Schließen einer Dialogbox vereinigt, wobei nochmals ein Form_center-Aufruf benutzt wird, für den Fall, daß die Big-Koordinaten inzwischen verlorengegangen sind. Natürlich könnte man sie auch zwischenspeichern, aber das würde bei Verwendung mehrerer

Dialogboxen zuviel Aufwand bedeuten, und außerdem ist diese Methode sicherer und universeller. Bei den Form_dial-Aufrufen kann ebenfalls wieder einer weggelassen werden (Flag%=2). Ansonsten werden als Parameter wieder nur die Baumadresse und die Littl-Koordinaten übergeben.

Undo_objc — Objektstatusflag löschen

Undo_objc(Baumadresse%,Obj_index%, maske%)

Auf diese Routine wird in der näöchsten Ausgabe noch ausführlich eingegangen. Hier sei nur soviel gesagt, daß sie in unserem Falle zum Deselektieren der Buttons benutzt wird. Dazu muß der Parameter den Wert 1 haben. Weitere Parameter sind wieder die Baumadresse und der Objektindex.

Rsrc_free — Freigeben des Resourcespeicherplatzes

Rsrc_free

Diese Routine ist endlich mal wieder eine reguläre AES-Routine, nachdem Sie mit unseren eigenen nun eine ganze Zeit lang vorliebnehmen mußten. Rsrc_free wird allzu gerne am Ende des Programmes vergessen. Durch sie wird der durch das Resource belegte Speicherplatz wieder freigegeben. Vergißt man sie, kann es nach mehreren Programmaufrufen zu einem „Out of memory" (kein Speicherplatz mehr vorhanden) kommen, da jedsmal erneut Speicherplatz beim Laden der Resourcen reserviert wird. Allein deswegen empfiehlt sich beim Rsrc_load-Aufruf die Fehlerabfrage.

Nun, die TEDINFO ist wiederum eine Tabelle, die aus 13 Worten bzw. 26 Byte besteht. In ihr sind alle wichtigen Daten für eine Textein- und ausgabe definiert. Um den Textstring aus einem EDIT-Feld zu bekommen, benötigt man die ersten vier Bytes (te_ptext) der TEDINFO. Diese enthalten die Adresse des schon lang ersehnten Textstrings, auf den wir schritt- bzw. zeigerweise zugearbeitet haben. Die Adresse wird dann von der Prozedur zurückgegeben.

Cursor am Textanfang

Zusätzlich haben wir in die Ptadr-Routine noch die Möglichkeit eingebaut, ein Textfeldeintrag zu löschen und somit den Cursor an den Anfang des EDIT-Feldes zu setzen (siehe ST-Ecke März '87). Gesteuert wird dies mit dem Parameter Flag%. Hat er den Wert 1, so wird das Textfeld gelöscht, und der Cursor steht am Anfang. Alle anderen Werte bewirken, daß ein vorhandener Text erhalten bleibt. So kann man zum Beispiel im Resource Construction Set schon einen Text vorgeben, der dann nur nach Bedarf entweder im Programm oder mit der Escape- bzw. Backspace-Taste gelöscht wird.

Get_dialogtext — Auslesen des Textstrings

Get_dialogtext(Textadresse%,*Text%)

Auch diese Routine ist selbst geschrieben und dient lediglich dazu, den Textstring aus dem Speicher auszulesen. Übergeben wird die Textanfangsadresse, die man zuvor durch die Ptadr-Routine erhalten hat. In einer Schleife wird dann Zeichen für Zeichen ausgelesen und zu einem Textstring zusammengefügt. Da man durch den PEEK-Befehl lediglich ASCII-Werte erhält, müssen diese durch CHR$ beim Zusammenfügen in Buchstaben konvertiert werden.

Die Schleife läuft solange, bis sie den Wert 0 ausliest, der immer am Ende eines Textfeldes steht. Zurück erhält man den fertigen Textstring zur weiteren Verwendung.

Box_draw — Zeichnen der Dialogbox (Zusammenfassung)

Box_draw(Baumadresse%,X%,Y%,W%,H%)

In dieser Routine werden alle drei bzw. vier Prozeduraufrufe zusammengefaßt, die man zum Zeichnen einer Dialogbox benötigt. Nach Belieben kann dabei der Form_dial-Aufruf mit Flag%=1 aus oben beschriebenen Gründen weggelassen werden. Man spart also Platz und gewinnt Übersicht, wenn man statt der einzelnen Aufrufe nur noch diese Routine aufruft.

Als Parameter werden die Baumadresse und die und die Littl-Koordinaten für die Form_dial-Routine benötigt.

Zum Programm

Das Programm selbst ist, wie man sieht, gar nicht so lang. Den Löwenanteil machen die Variablendefinition am Anfang und die eigentlichenProzeduren am Ende aus. Nach dem Start erscheint zuerst die erste Box, in die man ein Datum eingeben kann. Anschließend erscheint nach dem Anklicken des Buttons „weiter" die zweite Box, die zwei EDIT-Felder enthält. Hier lassen sich zwei beliebige Texte eingeben. Als Ausgang aus der Box gibt es hier die Möglichkeiten des Zurückverzweigens in die erste Box (Ex_obj%= Dialog1) oder des Beendens des Programmes (Ex_obj%=Bye).

Wichtig ist das Deselektieren des Buttons, mit dem man die Dialogbox verlassen hat, da dieser sonst bei einem erneuten Aufruf noch selektiert ist und invers (also schwarzer Hintergrund) mit weißer Schrift) dargestellt ist. Am besten deselektiert man gleich nach dem Schließen der Dialogbox automatisch den Button, da man es so nicht vergessen kann.

Nach dieser ersten Einführung werden wir uns im nächsten Heft näher mit der Manipulation des Objektstatus und der Objektflags beschäftigen. Eine Routine (Undo_objc) wurde ja bereits diesmal verwendet, aber noch nicht näher erklärt. Probieren Sie ruhig ein bißchen mit den vorhandenen Routinen herum, indem Sie eigene neue Dialogboxen entwerfen und programmieren. Sie werden sehen, wie leicht man nun Eingabemasken etc. erstellen kann, ohne eine aufwendige Eingaberoutine schreiben zu müssen, da man ja schon im RCS die Länge der Textstrings und die Art der Zeichen begrenzen kann.

(HE)





können ab sofort Ihren ATARI ST ohne Computerkenntnisse geschäftlich einsetzen! Komfortable Bedienung, Beratung und problemlose Handhabung garantieren Ihnen die optimale Ausnutzung Ihrer Hard- u. Software.

STS FIBU

STS FIBU ermöglicht professionelle Finanzbuchhaltung mit Ihrem ATARI. Sehr einfache Bedienung, auch durch den Computerlaien, und Übersichtlichkeit zeichnen dieses Programm aus.

Data Expert

Das Expertensystem. Künstliche Intelligenz in der Datenverwaltung.
Anwendungsbereiche: Geschäfts-, Hobby-, Privatbereich.
Einsatzmöglichkeiten: unerschöpflich.

FAKT

Die Fakturierung, die auch Ihnen ein unproblematisches Arbeiten ermöglicht. Zwei Adressen pro Kunde, mehrere Preise pro Produkt, individuelle Preisänderungen etc. machen dieses Programm sehr flexibel und somit universell einsetzbar. FAKT ist selbständig lauffähig, kann aber mit ADRESS und LAGER zu einem Komplettsystem ausgebaut werden.

DECIDE

Entscheidungen zu treffen durch künstliche Intelligenz. DECIDE betrachtet alle Faktoren objektiv und trifft immer die richtige Auswahl. Das Programm hilft Ihnen nicht nur Geld sparen.

LAGER

LAGER verwaltet neben Ihren Artikeln auch Ihre Lieferanten. Artikellisten, Lagerbestand überprüfen, Inventurprotokoll sind bei diesem Programm selbstverständlich. Sehr einfach anwendbar.
Ideale Erweiterung zu FAKT.

ADRESS

ADRESS ersetzt Karteikarten und verschont Sie vor lästiger und zeitaufwendiger Auswert- u. Sortierarbeiten. Mit Adress sind Listen und Etiketten blitzschnell erstellt.

STS FIBU	DM 498,--	ADRESS	DM 99,--
LAGER	DM 198,–	DataExpert	DM 99,--
FAKT	DM 198,–	DECIDE	DM 99,–

Sämtliche Preise sind empfohlene Verkaufspreise.

Unsere Produkte erhalten Sie bei verschiedenen Fachhändlern und direkt bei uns.

Wir senden Ihnen auch gern kostenlos und unverbindlich ausführliches Informationsmaterial zu !

Direktbestellungen richten Sie bitte an: **STS SOFTWARE - Stefan Seucan - Postfach 24 44 - 8600 Bamberg 1**
Telefon: 09542 - 83 48
Die Lieferung erfolgt per Vorkasse (V-Scheck) oder Nachnahme. Versandkostenpauschale DM 5,--

COMPUTERVERSAND WITTICH
Tulpenstr. 16 · 8423 Abensberg
☎ **0 94 43 / 453**

Atari 520 STM	588,–	Kabel u. Stecker	a. A.
Atari 1040 STF	1299,–	Atari ST mit 2,5 Megabyte	a. A.
Atari SH 204	1198,–	1st Teacher	99,–
Aufrüstung auf 1 MB	198,–	**Drucker**	
Monitor SM 124	444,–	Seikosha SP180A	499,–
Original Maus	98,–	Seikosha SP1200AI	598,–
Disk. Station SF 354	198,–	Seikosha SC80AI	1198,–
NEC Disk. Stat. 720 KB	398,–		

CompWare Büroelektronik GmbH

- **TKD 3,5" Markendisketten MF2 DD**
  100 % error free

20	50	100
95,–	225,–	420,–

- **Star NL 10 mit Interface 696,– DM**
  (Versand: unfrei UPS/Nachnahme)

- Bensheimerstr. 7, 6084 Gernsheim · Tel. 0 62 58 - 5 16 16
- Ernst Ludwig Str. 7, 6840 Lampertheim · Tel. 0 62 06 - 5 48 88

JETZT AUF DISKETTE

Das Siegerprogramm unseres Musikwettbewerbes

Komponieren Sie Ihr eigenes Musikstück mit dem **MUSIK-EDITOR**

Hau rein Zottl
nur 44,– DM
zuzügl. 5,– Versandk.
unverb. empf. Verkaufspreis



DM 44,–

- Sie erhalten eine Diskette samt deutscher Anleitung.
- Auf der Diskette befinden sich neben dem Musik-Editor noch einige Musikstücke.
- Bei Bestellung verwenden Sie bitte die im Heft beigefügte Buch- u. Software-Bestellkarte.

Bestellungen bei Ihrem ATARI-Händler (s. Einkaufsführer) oder direkt beim Verlag

Heim-Verlag · Heidelberger Landstr. 194 · 6100 Darmstadt-Eberstadt · Tel. 0 61 51 / 5 60 57

Kleinanzeigen

BIETE SOFTWARE

Orig. **Lattice C** T. 05251/56901

1st-Word+ VB 150,– 02833/7374

SIGNUM, Print Master, Ninja usw. (orig.) Tel. 06121/301904

■AKTIEN-Depotverwaltung: 79,–
■INFO: 07131/72639 o. 81163

Namensdatei und Testauswertung f. programmierte Aufgaben, mit Bewertung aller Aufgaben. Dazu div. Schulprogramme gegen 40 DM Vorauskasse bei Helmut Geiger, Wiesenstr. 13, 7250 Leonberg 7

+ + + + + **PD-Software** + + + + +
Freie Programmzusammenstellung gegen Unkostenbeitrag von DM 5,– pro 350 KB abzugeben. Gratisliste von: D. Metz; Dyckburgstr. 18; 4400 Münster.

ST-Uhrmodul, noch Garantie 60,– Hamlet Schachprg. 80,– diverse ST-Fachbücher DM 10–20. G-Data Utilities 25,–. 0221-519950

UCSD-P-System mit Basic, Pascal und Fortran für Atari ST wegen Systemumstellung zum Preis von nur 600,– DM zu verkaufen.

Einkommen-/Lohnsteuer 86/87 Neu: Gem/Ergebnis speichern/ Was wäre Wenn/30 DM Vork. einschl. Disk + Porto Jochen Höfer Grunewald 2a/ 021923368 5272 Wipperfürth

PD-Software
Zusammenstellung der Programme auf Diskette ist frei wählbar. Dadurch effiziente Bestellung. Gratisinfo bei Klaus Galz in 78 Freiburg, Sonnenbergstr. 55.

Etiketten/Labels f. a. Drucker Text+Bilder mischb. u. GEM 79,- **Spielautomat DISC Super Grafik dreh. Scheiben** 49,- ★ Info 5,- ★Demo (beide) 20,- ★ F. Walter Königsberger Str. 2, 3170 Gifhorn

ATARI ST - LOTTOSTATISTIK für 39 DM, 20 seit. Dokumentat. vorab gegen 2,40 in Briefmarken F. Klein, Dessauer Str. 5a, 89 Augsburg 21, Tel. 0821/812157

DIN 4701 Wärmebedarf + K-Zahl Pulldown-Menüs + GEM-Technik Programm DM 110; Demodisk DM 10 nur Vorkasse von J. Binder Eichendorffstr. 15, 5030 Hürth

Wg. Systemaufgabe: 1.Word+ neu DM 150,- Hanse, ST Karate, Karatekid2, Starglider, Twoontwo, Leaderboard - je DM 50,- T 0931/83509

Atari ST div. Originalprogramme Tel. 05171/3723

Public Domain Software
Incl. Diskette 5,50 DM
Atari Computer Club Reutlingen
Tel. 07129/2186 ab 18.00 Uhr

★ ★ **1st Proportional V2.02** ★ ★
Proportionaldruck im Blocksatz m. 1stWord/Plus incl. Fußnoten, Grafik (spez. Grafiktr. f. 24 Nadeldrucker!), versch. Zeilenlineale in PS/Blocksatz, Verkettung v. bis zu 25 Texten im Ausdruck mit autom. Seitennummerierung, usw.. Für alle PS-fähigen Drucker w. NEC P6/7, HR15, EM701, EM811, Gabi 9009 u.a. PRG(GEM) m. ausf. (60 S.) deut. Handb. **89 DM**
(Tel: 0241/54734)
KnIß, 5100 Aachen, Viktoriastr. 9

Von Lehrer für Lehrer! Notenverw. **Markbook plus** für DM 60,- 10 versch. Funkt., **GEM.** Handbuch für DM 5,- (bei Kauf Verrechng.) o. Info geg. Fr.umschlag (50 Pfg) H. J. Merkel/Nahestr. 28/6600 SBR

Verk. orig. Kalkumat+Datamat ST 150+50 DM (zus.180) 0201-7100425

Signum 330,- **SM Rechnung, Kunde Lager** 180,-, Orig., 07141-603702

Die neuen Vokabelprogramme:
FREE-Soft: **VocTeacher** : 15 DM
VocTeacher, V: 2,0 : 29 DM
VocMaster, V: 2,1 : 49 DM
Reiner Kocher, T: 089/3134946
Caracciolastraße 16, München 45

DFÜ ST (autom. Wählen, Logon, Scanner, bis 1200/75 Bd 58 DM
Fixsternhimmel (Nachführung, Suchen, editierbar u.s.w. 79 DM
Wordcheck für 1stWord, SM-Text u.a. (mit Lexikon) 69 DM
Ligatabelle (komfort.) 39 DM
Ahnenforschung 59 DM
Katalog kostenlos. RJR-Software Buchenweg 2, 5172 Linnich 6

★ ★ **PD-Service DM 5,00** ★ ★
Angebot wie PD dieser Ausgabe Einzeldiskette SS **DM 5,00**
Pakete 1–10, 11–20, 21–30, 31–40 und 41–50 je **DM 38,00**
auf je 5 Disketten DS
Preis inkl. Disk zzgl. Porto und Verp. DM 5,00 (Ausl. DM 10,00) NN plus 1,70 besser V-Scheck
I K S , Enzianstr. 18
7500 Karlsruhe 41
ab 18 Uhr 0721/473644

Kleine Anzeigen – kleine Preise
ST-Software, zum Beispiel:
Flightsimulator II DM 107,–
Sundog DM 87,– und alle anderen.
Gratisliste bei A. Gauger Software Buhlstraße 16a, 7505 Ettlingen
Telefon: 07243/78511 o. 16959

MW-C Compiler, Version 1.0+**2.0** mit Büchern 300,– Mercenary 30 DM, Uwe ab 18⁰⁰, 0711/344314

Wärmebedarfs./KZahlberechnung
DIN4701/83 GEM Bedienerführung
Dipl.Ing. V. Koch, Am Mehnacker 11
3563 Dautphetal 3, Tel. 064687652

Wir haben die Hardware...
Wir haben die Software...
Wir haben das Zubehör...

Schreiben Sie uns.
Wir antworten sofort.

Gerald Köhler
Soft- und Hardware Vertrieb
Mühlgasse 6
6991 Igersheim
Tel. 07931/44661 (24h-Service)

■**ST VOICE** Sprachausgabe auf■
■dem ATARI ST. Hardwareer-■
■weiterung incl. Treiber-Software.■
■**ST SCHECK** GEM-Programm■
■zum Verwalten und Drucken■
■von Schecks und Überweisungs-■
■formularen. ■
■SCHLEGEL Datentechnik■
■Schwarzachstr. 3, 7940 Riedlingen■
■ Tel. 07371/2317, 2438 ■

STATISTIK deskrip. Daten, para /nonparametr. Tests, Korrelationstabellen etc. fertig zur Übernahme in Textprg.(!), auf Drucker DM 149,- Dipl. Psych. M. PRALL, Isestr. 57, 2 Hamburg 13

Schon ab **6,50 DM** gibts PD-Software auf einer **Maxell Qualitätsdisk.** Suchen Sie aus **über 200** Disks **Ihre** heraus! Gratisinfo bei **Ralf Markert, Balbachtalstr. 71, 6970 Lauda, Tel. 09343-8269**

An alle ST-Fans!
Über 150(!) Disketten PD-Soft, auch doppelseitig, 200 Profi-Programme und viel Zubehör zu interessanten Preisen. Wir sind die ST-Spezialisten!
Gerald Köhler
Soft- und Hardware Vertrieb
Mühlgasse 6 ★ ★ **6991 Igersheim**
Tel. 07931/44661 (24h-Service)

99 Pfennig für eine PD-Disk, gratis ist mein Katalog und weitere Preishämmer finden Sie bei Ralf Markert, Balbachstr. 71, 6970 Lauda, Tel. 09343-8269

PD-Software zu Superpreisen!!
PD-Disk 5,50 DM freie Wahl der Prog. DM 7 inkl. Disk. **Mehr** KB für Ihr Geld also **Gratisinfo** von **EUSAX** Haus Gravenerstr. 85, 4018 Langenfeld – es lohnt sich

Erweitere deutsches Wörterbuch f. 1st Word Plus auf ca. 145 KB, 25,– DM. **CRYPT-ST,** ver- u. entschlüsselt Texte von 1st Word bis 32K (f. S/W Monitor) 30,– DM Alle Preise **zuzügl.** Porto/Verp. Peter Schmidt
Kuthsweg 45, 4000 Düsseldorf 1

An alle ST-Fans!
über 150(!) Disketten PD-Soft, auch doppelseitig, 200 Profi-Programme und viel Zubehör zu interessanten Preisen. Alle Atari-Computer deutlich unter Listenpreisen und mit einem vollen Jahr Garantie.
Wir sind die ST-Spezialisten!!
Gerald Köhler
Soft- und Hardware Vertrieb
Mühlgasse 6 ★ ★ **6991 Igersheim**
Tel. 07931/44661 (24h-Service)

60.000.000 Bytes PD-Software Spiele, Anwendungen aller Art, Grafik- und Musikprg's, viele Demos die zeigen was Ihr ST so alles kann. Sie werden überrascht sein. Ausführliche Beschreibungen über unser PD-Angebot mit weit über 1000 Programmen auf 150 Disketten gibt es bei:
Gerald Köhler
Mühlgasse 6
6991 Igersheim
Tel. 07931/44661 (24h-Service)

Public-Domain-Blitzversandt:
Wir bieten über 150 Disketten für Ihren Atari ST! Preise:
DM 6,– für einseitige Disketten
DM 8,– für doppelseitige
Inclusive Diskette!
Gratisliste anfordern bei:
A. Gauber Software, Buhlstraße 16a
7505 Ettlingen, 07243/78511

★ ★ ★ Public Domain ★ ★ ★
Neu, aber mit großem Angebot. Viele Demos von prof. Software Super Preise. Info bei:
Ulrike Nolte, Wasenweilerstr. 11a
7817 Ihringen, Tel: 07668-7301

PD-Software ist unser Spezialgebiet. Seit über einem Jahr. Schreiben Sie uns. Es lohnt sich:
Gerald Köhler
Mühlgasse 6
6991 Igersheim
Tel. 07931/44661 (24h-Service)

★ ★ACHTUNG LEHRER!★ ★
„Teacher's Friend" ist da.
Die komfortable Notenverwaltung. nur DM 69,– direkt bestellen oder Info anfordern bei U. Kunz Junge Hälden 3, 75 Karlsruhe 41

★ ★ **PD-SOFTWARE** ★ ★
Spiele, Musik, Anwendungen, etc auch die PD v. ST Computer
Superpreise!
Gratisinfo bei E. Twardoch, Kaiser-Wilhelm-Str. 88, 1000 Berlin 46

PD-Software ist unser Spezialgebiet. Seit über einem Jahr. Schreiben Sie uns. Es lohnt sich:
Gerald Köhler
Mühlgasse 6
6991 Igersheim
Tel. 07931/44661 (24h-Service)

1. Fußball-Bundesliga-Verw. mit Erg. u. Tab. seit 63; Info 2D Mjnur s/w; HJR Software Evolution, Warendorfer Str. 4, 4722 Ennigerloh 2; Preis: 69 DM (NN)

ATARI-ST-Software in großer Auswahl ab Lager lieferbar. Kostenlose Preisliste anfordern. HEINICKE-ELECTRONIC Kommenderiestr. 120 4500 Osnabrück Tel. 0541/82799, Tx 944 966

ST Profi-Sopftw. Freak-Preisen VIP(GEM) 428,– dBMan 3.0 338,– Flugsim. II 98,– Schach 49,– **★A. v. Zitzewitz★** 089-584405 Perhamerstr. 70 · 8 München 21

Stabwerk- u.v.a. Statik-Progr. sowie **über 60 Mathem.-Progr.** Info (1 DM) bei Dipl. Ing. Jürgen Bullmann, Zur Hindenburgschleuse 3, 3000 Hannover 71

BAUSTATIK: Holz-Beton-Stahl- **AUSSCHREIBUNG:** Angebot-Massen-Programmservice Schmidt Burgstr. 4, 5376 Marmagen Tel. 02486/7417

BIETE HARDWARE

ATARI-Echtzeituhr akkugepuffert Einbau ohne Löten DM 89 Romport nicht belegt T. 02337/1239

SF 354 wenig ben. 089/8112745

SF354 200,- UWE 18⁰⁰ 0711/344314

Floppy-Modelle 3.5 Zoll:
ST- 3 –> 1★720 KB: 369 DM
ST-13 –> 2★720 KB: 699 DM
Modelle mit 3.5 und 5.25:
ST-16 –> 2★720 KB: 868 DM
★★ 100 % ST-comp., Netzteil
★★ eingeb., mit NEC 1036 A
★★ komplett Anschluß-fertig
Weiteres: Alles Endpreise
––> NEC FD 1036A: 239 DM
––> SF354 Umbau mit 1036A
––––> nur 298 DM! <––––
––> Mega-Atari im PC-Gehäuse:
––> 1–4 MB, 2 NEC-Floppies,
––> SW-Monitor, Software!!
––––> ab 1599 DM <––––
FISCHER Computersysteme
Tel: 0 61 51 / 5 13 95

Doppelstation (NEC 1036) 695 DM Software zu Spitzenpreisen. Ataripoint – Rauch 07031-52607

Atari-Monitor-Umschaltgerät stab. Metallgehäuse + Tonausgang Sofortlieferung – DM 89 + NN T. 0303027965 tägl. 20–24 Uhr

Atari 520ST+/1MB;SF314;SM124; gegen Höchstgebot! T. Steinkuhl, Katenbäker Berg, 2878 Wildeshaus

SF354 1/2 J. 140,– 07941/35770

EPROMMER m. Textool nur 145,- Accu-Uhr nur 45,- DM 040-862771

★★We are the Champions★★ Spottbilliges Atari-Zubehör, über 200 PD-Disks, reihenweise Angebote und, und, und. Gratis-info bei **Fa. Ralf Markert, Balbachtalstr. 71, 6970 Lauda**

SF354 DM 200,– 02361/654151

SF354neu **150,-**+Softw. 02105/8967

SF354 neu 150,- Tel. 0711/573203

SF 314 DM 320,– 02677/1234

Doppel-Floppystation: NEC-1035 ST-mod. (neuwertig) VB: 650,- DM Tel: (09568) 2668

SF 314, 400,–, 07123/7376 Klaus

Farbmon. Orion CCM-1280 m. Anschlk.+Softw. 700,- 08231/4926

Atari 260ST 1 MB, Floppy SF354, Monitor SM124, Scart-Kabel, Drucker OKI 183, div. Software, (Textomat, Datamat), 7 Monate. Preis: VHB T. 0761-892806/abends

260ST, SF354, SM124 + Maus, 1MB RAM, 7 Mon., VB 1480, Tel: 07541/43466

260ST, SM124, SF354, Maus, Acustickoppler, Lattice C, Seka Ass., div. Bücher 2200,– DM Tel. ab 20 Uhr: 06101/1089

Farbmonitor Orion CCM 1280 NP > 900 DM für 598,- 09371/67155

Wegen Hobbyaufgabe zu verkaufen: Atari 1040 STF+SM124+Maus +Epson-Drucker LQ 2500 (24Nad.) +Software (u.a. BS-Handel) zus. für schlaffe 4.300,– (VB) Tel. Andreas 0211/232714

Brenne PRG's; TOS auf Eprom T. 05171/3723

SF 354 DM 100,– 0711/2626271

Extern steckbarer Schalter zw. SW u. Farbmon. (kein umstecken mehr) 47,90 DM. Floppy 720 KB anschlußf. 385 DM. 02305/74258

Thomson Farbmonitor 500,- DM. Drehfuß f. jeden Monitor 30 DM. Büro 07641/459277 M. Vetter L.-Jahnstr. 17, 7835 Tenningen 1

Thomson Farbmonitor RGB f. Atari 3 Mon. alt. 850 DM VB Tel. 02191/72164

Atari 1040 STF, Maus + Abdeckung (11 Monate alt) wegen MEGA-STs zu verk.: **VHB 1500,** 04321/37564

ROM-TOS 100 DM (VB) Tel. 02630/7525

Hobbyaufgabe: 1040STF+Monitor SM124 + Software + Bücher VB. DM 1550,–, Drucker FX80plus VB. DM 550,– Tel.: 089/7237830

RHOTHRON-Atari-PC-Gehäuse zu verkaufen Tel. 05251/27707

★★ GALACTIC-CHESS ★★
Schachprogramm 'Deep Thought'
Exklusiv von uns, nur 89 DM
Demoversion 0.9 (PD) nur 10 DM
TV/AV-Modulatoren ab 130 DM
1x3.5"-Floppy für ST 450 DM
2x3.5"-Floppy für ST 799 DM
NEC 1036A (3.5"-Floppy) 278 DM
Floppy-/Monitorstecker uva.
Info bei: Stachowiak-Computer
4300 Essen, Burggrafenstr. 88
Tel.: 0201/273290

ST-Video-Interface mit HF-Modulator und Anschl. für Fremdmonitor, incl. Zubehör, DM 128,– ST-Scart-Kabel, Tel. 06126/52919

SUCHE HARDWARE

ST-VOICE gesucht 0541-596539

SUCHE SOFTWARE

Suche Softw. jegl. Art + Kontakte A. Klabis: 040/7225893: ab 18 Uhr

Suche Handb. VIP (GEM) **dt.** tausche geg. Handb./Softw. 0201/7100425

Suche Salix- oder **MProlog** und Cambridge **Lisp** 05641/8754

Programm zur Verwaltung einer **Mineralien**sammlung **(Systematik)** gesucht. 07394/537 (abends)

Suche Fibu mit Bilanzerstellung + intellig. Datenbankprgr., E. Baumann, 2950 Leer-Loga, Am Julianenpark 9 A

für 520ST Bundesligatabprogramm. viel Auswertu. und Grafikübers. Glaw, In d. Breite 2, 7700 Singen

Suche **Orig. Campus Prg.** günstig 0911/505037 ab 18 Uhr

KONTAKTE

Wer hat Lust und Zeit, einen C-Programmier-Workshop mit zu organisieren, konzipieren, mitzumachen? Wer macht mit? Michael Ziegel, Wolfgangstr. 21, 8000 München 80, 089/4470027

Suche Kontakte + Tauschpartner G. Butkus Windelsbleicher Str. 91 48 Bielefeld 14, Tel. 0521/69689

★★ ST-Kontakte weltweit ★★
Swop-Shop verbreitet Ihre private Kleinanzeige **weltweit** und **kostenlos!**
Auch für private Verkäufe bzw. Kaufgesuche bestens geeignet. Telefonische Anzeigenannahme Deutschland 06151/58912
★★★★★★★★★★★★★★

WE WANT YOU!

Gesucht
fähige Programmierer
für Atari ST und Amiga
in
Assembler, C, Pascal, Basic
für
Anwendersoftware, Systemlösungen
und Spiele.
Bevorzugt PLZ 8
Info bei:
S.- O. Hanssen
Hunoldsgraben 11
8900 Augsburg 1

TAUSCH

Tausche PD-**Astronomie** Software Daniel Roth, Brücker Mauspfad 448 ★★★ 5000 Köln 91 ★★★

VERSCHIEDENES

Suche für meinen NEC P6 eine Kunstlederhaube. U. einen Manuskripthalter. G. Butkus. 0521/69689

Soft- und Hardware für Atari ST Katalog bei Soft- und Hardwarevertrieb Günter Stahl, 4100 Duisburg 26, Pf. 260167

Disketten 3,5" MF2DD Panasonic 10 Stk. 55 DM frei Haus per NN oder V-Scheck, K&L, Dießelweg 52, 2900 Oldenburg, T. 0441/203539

SCHWEIZ – TIEFSTPREISE auf allen ATARI + AMIGA Computern, inkl. Peripherie + Software!!! by COMPU-TRADE Emmenstr. 16, CH-3415 Hasle-Rüegsau (034) 614593 auch abends

Suche Beschreibung Flight Sim. + Signum für ST. 06102-36896

Anzeigenschluß für Ihre Kleinanzeige in Heft 7/8 1987 ist der 25. Mai 1987.

Privatanzeigen kosten pro Zeile und Spalte DM 7,– incl. MwSt.
Gewerbliche Kleinanzeigen kosten pro Zeile und Spalte DM 15,– zuzügl. MwSt.
Zur Einsendung bitte die beiliegende Karte verwenden.
Bearbeitung nur gegen Vorausscheck (keine Überweisung).

Die Neuheiten des Monats

Auch in diesem Monat haben wieder zahlreiche Programmierer ihre Programme unserer Public-Domain Sammlung zur Verfügung gestellt. Ihnen wollen wir an dieser Stelle im Namen all derer, die mit dieser Software arbeiten, danken.
Diesen Monat können wir Ihnen unter anderem einige schöne Spiele bieten. Zum einen MAZIACS, bei dem eine tapfere, unternehmungsfrohe Figur auf dem Weg ist, eine Truhe mit Gold zu finden. Doch das ist bei weitem nicht das einzige neue Spiel. Die Freunde der MIDE-Szene erhalten zwei Programme, um Ihren Synthesizer mit den ST zu koppeln. Dabei ist eines speziell für den Schlagzeugsound.

Zum Zeichnen von Schaltplänen, was per Hand doch sehr mühselig ist, können wir Ihnen E-Plan empfehlen; es wurde zu diesem Zweck geschrieben. Doch, schauen Sie sich den Inhalt der PD-Sammlung einfach an.

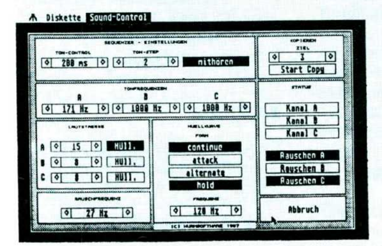
Sämtliche Disketten werden weiterhin auf Qualitätsdisketten der Marke **FUJI MF1DD (135 Tpi)** geliefert, da wir damit, ein- und doppelseitig, die besten Erfahrungen sammelten. Wie und wo sie zu erhalten sind, erfahren Sie auf der nächsten Seite.

Wollen auch Sie sich mit Ihren eigenen Programmen an dieser Sammlung beteiligen, so schicken Sie sie einfach direkt an die ST-Computer Redaktion · Schwalbacherstraße 64 · 6236 Eschborn.

Wir freuen uns über Ihre Beteiligung an dieser nichtkommerziellen Sammlung.



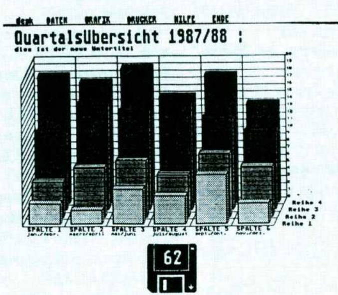
- *Synthi:* Ermöglicht bestimmte Sounds zu erzeugen. Dabei werden alle Feinheiten des Soundchips ausgereizt. Der fertige Sound ist auf Diskette speicherbar und damit in jedes eigene Programm einzubinden. Der Aufruf des Sounds erfolg tmit der interuptgesteuerten DOSOUND-Routine. Das Programm wurde in ST-Pascal erstellt, der Source-Code ist beigefügt.



- *Waschi:* Ein Hilfsprogramm für den unerfahrenen Junggesellen. Hilft bei der Wäsche und berät bei der Wahl der Waschprogramme.
- *Expert:* Ein einfaches lernbegieriges Expertensystem.
- *Adressen:* Adressveraltungsprogramm mit Etikettendruck und einfacher Textverarbeitung



- *Hacomini:* Waren Ihnen Ihre Hardcopies zu groß, so ist dieses Programm genau das Richtige. Hacomini erzeugt eine Hardcopy mit kleinsten Maßen (48 mm ★ 68 mm).
- *Helper:* Eine Mischung aus CLI (command line interpreter), Diskmonitor und Druckerutility. Z. B.: Sektor Editieren, Statusregister, Resource anzeigen, Druckerausgabe, Bilder anzeigen, Diashow, Tastaturkonfiguration, Dateiattribute ändern, etc...
- *Bidipo:* Kleines Programm mit großer Wirkung. Ermöglicht das Zusammenfügen mehrerer Bilder beliebigen Formats zu einem langen BAnd. Dieses wird zyklisch über den Bildschirm gescrollt. Ideal für eigene Diashows oder Werbegrafiken.
- *Direct all:* Ausdruck des gesamten, oder auch nur ausschnittsweisen Directories auf dem Drucker.
- *Überweisung:* Das Überweisungsdruckprogramm, das auch in dieser Ausgabe abgedruckt wurde.
- *Datobert:* Stellt Werte dar und zwar als Linien- und Kuchengrafik, sowie in dreidimensionaler Balken- und Flächengrafik. Dabei werden die einzelnen Datenserien überlagert und können somit leicht verglichen werden. (s/w).





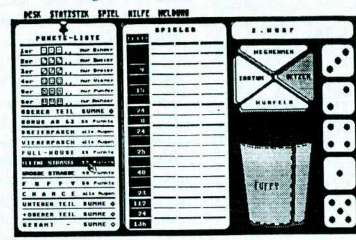
MIDI Diskette

- *MIDIDRUM:* Rhythmusmaschine mit leicht und vielseitig editierbaren Sounds.
- *MIDIPLAY:* Erlaubt das Einspielen, Speichern und Ausgeben von Midi-Sequenzen, incl. Source-Code in GFA-Basic.

63

Spielesammlung

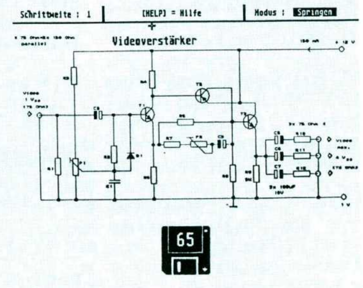
- *FUFFY:* Sehr schön gestaltetes Kniffel-Spiel mit zusätzlicher Auswertegrafik. Die gefallenen Würfel und der statistische Spielstand zwischen den Mitspielern werden dreidimensional sichtbar gemacht (s/w).



- *Snake:* Eine Schlange schlängelt sich durchs Leben, auf der Suche nach Fressen. Aber manchmal beißt sie sich auch in den eigenen Schwanz oder stößt gegen eine Wand, was ihr nicht gut bekommt.
- *Bomber:* Bei diesem Spiel muß man mit einem Flieger in eine Höhle vordringen und alles abschießen, was sich bewegt – sehr einfältig. (f)
- *Reversi:* Umsetzung des bekannten Spiels auf den ST. (f)



- *E-Plan:* Konstruktionsprogramm für elektronische Schaltungen. Alle elektronischen Bauteile erscheinen per Knopfdruck auf dem Bildschirm. Batterieanschluß, Birnchen, Kondensator, Diode, Elko, Lötfahnen, Gleichrichter, IC's, Lautsprecher, Induktivität, Masse, Lötpunkt, Widerstand, Schalter, Transistor, Op-Verstärker, Diac, Thyristor, Motor, FET, u.a. Außerdem können die Schaltung beschriftet und die Bauteile in die Stückliste übernommen werden. (s/w)



65

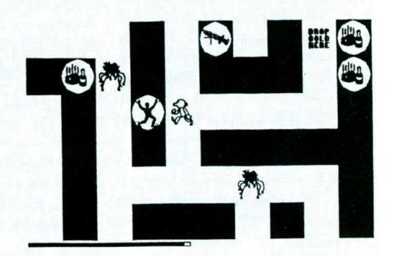
Mal- und Zeichenprogramme

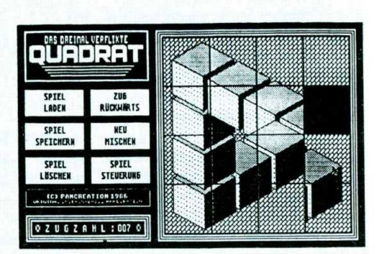
- *Columbia:* Malprogramm mit vielen Extras. Mehrere Bildschirme, Blockoperationen, Spiegeln, Verzerren, Drehen, Lupe. (s/w)
- *Paintlux:* Malprogramm mit zusätzlicher Bibliotheksverwaltung. Dadurch können z. B. elektronische Bauteile aus der Bibliothek in das Bild eingefügt werden. (s/w)



Spieledisk mit Source-Code in OMIKRON Basic

- *MAZIACS:* Abenteuerspiel. Auf der Suche nach einer Goldtruhe muß unser Held einige Hindernisse überwinden. Im Labyrinth warten bösartige Spinnen, mit denen man aber, je nach Kondition und Bewaffnung, fertig wird. (s/w)





- *Quadrat:* Es gilt ein verworrenes Bild zu ordnen. Grafische Umsetzung eines bekannten Puzzlespiels. (s/w)



Public-Domain Service

Die auf diesen beiden Seiten aufgeführten Disketten können gemäß untenstehenden Versandbedingungen direkt über den ST-Computer Public-Domain Service bezogen werden. Die Programme werden auf sehr guten Disketten der Marke FUJI-FILM MF1DD (135 TPI) geliefert, mit denen wir gute Erfahrungen sammelten und die sich problemlos ein- und zweiseitig nutzen lassen.
Pro Diskette entfällt ein Unkostenbeitrag von DM 10,–

1. Diskette
● *Neochrome V.06:* Malprogramm [f] ● *Doodle:* Malprogramm [s/w]

2. Diskette
● *Joshua:* Diskmonitor – CP/M Emulator (CP/M 80) ● *Megaroids:* Weltraumspiel ● *Diskformatter:* erweitert Diskkapazität auf 399 K bzw. 809 K ● *Kartei-Kasten:* luxuriöses Karteiprogramm ● *50/60 Hz*-Umschaltung für Monitore.

Diskette 3 und 4 mit Volksforth 83 wurden durch die Disketten 49, 50, 51 ersetzt.

5. Diskette
● *Ramdisk* mit Source-Code ● *SG 10* Druckertreiber als Desk-Accessory ● *JAM Term:* Terminalprogramm ● *Diskcopy* ● *ST Grafic:* bewegte Grafiken in 3D ● *CP/M Emulator V 8.4:* neuste Version

6. Diskette
Diese Diskette beinhaltet einige Nutzprogramme für den CP/M-Emulator. Lauffähig mit Version 8.4.
● *CPMTOS:* TOSCPM wandelt CP/M-ASCII-Files in TOS-Format und zurück ● *SCOPY:* Kopierprogramm für ein Laufwerk ● *DDTZ:* Debugger ● *SETKEY:* Tastaturumbelegung, u. v. m.!!

7. Diskette
● *XLISP 1.71* der Interpreter zum ST-Kurs Künstliche Intelligenz

8. Diskette
● *Diskmon:* ● *GfA Basic-Demo:* ● *Kuvert:* zum Bedrucken von Kuverts ● *Mini-CAD:* ein praktisches Programm mit Source-Code ● *Vokabel:* Vokabeltrainer ● *Superformat:* unterschiedliche Diskformate

9. Diskette
● *Funkplot:* Funktionsplotprogramm incl. Funktionseingabe über Tastatur, Integration und Differenzation ● *Konvert:* Konvertierung von Farb- auf Monochrombilder in 3,6 Sekunden ● *Spriteed:* ein kleiner Spriteeditor ● *Super:* ein Superhirnspiel [f]

10. Diskette
● *Iconeditor* ● *Puzzlepuzzle:* Puzzleprogramm ● *Krabat-Schach* Schach für den ST

11. Diskette
● *PROLOG 10* Die Sprache der neuen Generation. Begleitsoftware zum Kurs 'künstliche Intelligenz' incl. 95 KByte Anleitung.

12. Diskette
● *Rechner:* ein luxuriöser Taschenrechner ● *Calc:* Taschenrechner als Desk-Accessory ● *Biorhythmus:* Erläuterung überflüssig

13. Diskette
● *Pascalshell:* GEM-Menüsteuerung für ST-Pascal. ● *MMCOPY .ACC:* Filecopy als Accessoir ● *Happy 4:*'Vier gewinnt' ● *Labyrint. ACC:* Labyrint als Accessoir ● *MAD . ACC:* Ihr Rechner entwickelt Eigenleben ● u. a.

14. Diskette
● *Tempelmon:* Monitor von Th. Tempelmann [s/w] ● *Transistorvergleicher:* Transistordatenverwaltung ● *Bitte ein Bit:* Bitcopyprogramm ● *Diskmon* ● *VIFibu:* Kleine Finanzbuchhaltung ● *EX 06:* Fängt Systemfehler und zeigt sie an.

15. Diskette
Druckeranpassung 1st_Word
Auf dieser Diskette befinden sich alle uns zur Verfügung stehenden Druckertreiber. Es sind folgende Treiber enthalten:

ASCII	Epson RS80	Epson_12
Epson_LX80	IBM-Komp	Gabi 9009
IBM	MP_165	KX-P1092G
LQ_800	NL10_IBM	NEC_P2&P3
NEC_P6&P7	OKI_M192	NL10_Para
OKI_M182	TA-SE320	QUME
SMM804	Corona D300	TAXAN 810ST
C 8028	Gabi 9009	Brother M 1109
Citizen MSP10	Olivetti ET 121	Olympia Carrera

16. Diskette
Sammlungen diverser Programme incl. Source Code [s/w] ● *Etikettendruck:* bedruckt Etiketten (Basic) ● *Funkplot:* mit Funktionseingabe (Basic) ● *Kassetten- und Videodatei* unter GEM (Basic) ● *Typewriter:* Schreibmaschinenkurs (Basic) ● *LIFE:* Fortpflanzungs-Rechnung (Pascal) ● *Schiffe:* Schiffe versenken (Demo für GEM-Routinen) (Basic)

17. Diskette
Programmsammlungsdiskette incl. Source in Basic [s/w] ● *Aktien:* Der Aktienmarkt ● *Habu:* Verwaltung von Konten ● *Lohnbuchhaltung & Versanddruck* ● *Steuer:* Einkommens- und Lohnsteuerberechnung.

18. Diskette
● *Doodle plus:* erweiterte Doodle Version; incl. Bilder ● *Power:* Sonderzeichensatz [s/w] ● *Konverter:* Wandelt alle Bildformate ● *Datum:* Datum auf Disk ● *Smooth:* Scrollt eine 64 000 KB Doodle-Bild ● *Format:* erzeugt diverse Diskformate ● *Kbug:* ST leidet unter Käferbefall

19. Diskette
● *Kermit*[plus]: Stark verbesserte GEM-Version des Terminalprogramms ● *Disksort:* Druckt Diskinhalt als Liste oder als Etikette ● *Terminal:* Terminalprogramm ● *Fontedit:* Zeichensatzeditor für GEM-Draw und Easy-Draw Fonts. [s/w] ● *OKI-Init.ACC:* Steuercodes für Microline 182 incl. Source-Code

20. Diskette
● *Harddisk:* Treiber für Harddisk. Automatischer Start von der Platte (keine Bootdiskette nötig) ● *Toram:* Kopiert Programme in Ramdisk ● *RSCAnalyse:* Analyse von RCS-Files [s/w] ● *DESK Uhr. ACC:* Zeigt ständig Zeit und Datum an. Drei einstellbare Weckzeiten. ● *GEM-TOOL:* Demo für GEM-Programmierung [s/w]

21. Diskette
● *Adress:* Private Adressverwaltung ● *Etikett:* Etikettendruck ● *Haushalt:* Privatkontenverwaltung ● *Selektor:* Dateiverwaltung (30 frei erstellbare Felder)

22. Diskette
● *Plot:* Grafische Darstellung von Wertepaaren ● *Myterm:* Terminalprogramm (GfA-Basic) ● *VKT:* Vokabeltrainer

23. Diskette
● *DGDB:* 'Das große deutsche Ballerspiel', Aktionspiel siehe RELAX 12/86 [f] ● *Kalah:* Das bekannte Brettspiel ● *Fractals:* Schnelle 'Apfelmännchen-Version' in Assembler.

24. Diskette
● *Taurus:* Strategiespiel für mehrere Spieler. Siehe Relax 12/86. Schöne Grafik, viele Optionen [f], (leider nur Mega-Rechner)

25. Diskette
● *Disk Engineer:* Diskutility (Rescratch, ...) ● *Omikron Basic:* Run-Only Interpreter des neuen Basics

26. Diskette
● *Codetabelle:* ASCII-Tabelle (Accessoir) [s/w] ● *Life_Game:* Das bekannten Generationenspiel (Accessoir) [s/w] ● *Term:* Und noch ein Terminalprogramm ● *Biorhythmus:* ST-Basic schlägt zurück [s/w] ● *Konto:* Kontoführung in GfA-Basic [s/w] ● *KX P1091:* Accessoir zur Druckersteuerung

27. Diskette
● *NEC-Treiber:*Hardcopy-Treiber für 24-Nadeldrucker (NEC und kompatible). Aufruf über ALT+HELP ● *ST-Sound Demo:* Die Siegerdemo des ST-Computer-Sound-Wettbewerbs. 'Zottl, Knurp und Schnatter hauen rein' (siehe Titelbild ST-Computer Nr. 12)

28. Diskette
● *GfA Basic Programme incl. Source* ● *Funkplot:* [s/w] ● *Terminal:* Terminalprogramm in GfA [s/w] ● *Konverter:* Konvertiert ST-Basic in GfA-Basic ● *ST-Kill:* Utilityprogramme ● *Sprite-Editor:* [s/w]

29. Diskette
● *Diskutility:* Der Weg ans Directory (manipuliert Löschschutz, erzeugt waschechtes IBM-Format, und mehr) [incl. Source in Pascal] ● *Adress_pas:* Adressverwaltung, die einige käufliche Programme das Fürchten lehrt. ● *ST-Graph:* 3-D Balkengrafik [s/w] [incl. Source in GfA]

30. Diskette: Spielesammlung
● *Bohnen* ● *Schiffe:* Schiffe versenken ● *Mastermind:* Superhirn für Superhirne [f] ● *Dragon_hunt:* Ein äußerst gefräßiger Drachen will ausgerechnet Ihnen an den Kragen [f] ● *Memo:* Memory (2 Versionen) ● *Solitaire:* Das bekannte 'Taschenspiel' ● *4 Gewinnt:* Spiel in GfA-Basic incl. Source [s/w]

31. Diskette: Spielesammlung
● *Patience:* Wunderschöne Version dieses Kartenspiels (siehe RELAX Nr. 1/87 [s/w] ● *3D-Irrgarten:* Finden Sie einen Weg aus einem dreidimensionalen Labyrinth. Trotz ST-Basic sehr schnell [s/w] ● *Avalun:* Sie verwalten ein Königreich (ST-Basic)

32. Diskette
● *Mathlib:* Eine umfangreiche Mathematikbibliothek für Pascal. Trigonometrische, Matrizenalgebra und vieles mehr, direkt zum Einbinden ● *Finanzmathe:* Berechnung von Zinsen, Laufzeiten, Laufzeit, Tilgungsplänen, etc.

33. Diskette: diverse Utilities
● *Encrypt:* Ver- und Entschlüsseln von Dateien ● *Ramdisk:* Ramdisk beliebiger Größe ● *Recover:* Regeneriert gelöschte Files (!!) ● *Sterm:* VT52 Terminal + Source in Assembler ● *Freeram:* Zeigt freien Speicher

34. Diskette: Der Assembler zum Kurs
● *Assemblerpaket nach M 68000-Standard* incl. 2-Pass Assembler, Disassembler, Debugger, Editor.

35. Diskette
● *Andromeda:* Zeichenprogramm (DEGAS-ähnlich) ZOOM, Beschriften in vier Richtungen und allen Größen, Kopieren, Linienstil / Art / Breite und alle normalen Grafikbefehle [s/w] ● *3D-Konstruktion:* Zum Entwurf dreidimensionaler Gegenstände [s/w] ● *Globus:* Animierte Erdballgrafik

36. Diskette
● *CLI:* vollständiger Kommandointerpreter (CP/M, MS-DOS Oberfläche) ohne Maus. Ausführen von Batch-Files, Autostart, Kopieren auf Ramdisk, Belegung der Funktionstasten, Ausführen von Programmen (TOS + GEM), etc. ● *IBM. Drive:* Ermöglicht den Filetransfer IBM (–) ST ● *Drupatch:* Zum Ausdruck von Source Files. Mit Zeichenwandlung wie 1ST Word, universell anpaßbar. ● *DB_Etikett:* Druckt Etiketten aus DB-Master

37. Diskette
● *SCIENTFIC:* Zeichensatzeditor. Erzeugt Anpassungsfile für 1st Word [s/w] ● *Convert:* Konvertiert ST- und Microsoft Basic Programme in GfA-Basic ● *Check:* Rechtschreibkorrekturprogramm. Erstellt eigenes Lexikon, Übernahme der neuen Wörter, Korrektur direkt im Text.

38. Diskette
Alle Bilder des ST-Kalenderwettbewerbs, die in Heft 1/87 veröffentlicht wurden, incl. Picture-Compactor.

39. Diskette
● *Jackpot:* Einarmiger Bandit, incl. Sourcecode in GFA-Basic [s/w] ● *Lucky:* Spielautomat [s/w] ● *Hanoi:* Türme von Hanoi.

40. Diskette
● *Deluxe Fontmaster ST:* Druckprogramm zum Erstellen von Karten und Tabellen, mit vielen Zeichensätzen, Proportionalschrift und Blocksatz, eigener Text- und Zeichensatzeditor. Bis zu 10 Schriftarten in einem Text mischen und vieles mehr [s/w]

41. Diskette
● *Painter:* Zeichenprogramm unter GEM. Verfügt über einige Besonderheiten. Z. B. Zoom, Linien und Füllmustereditor, Blockoperationen (Verschieben, Drehen,...) [s/w] ● *VIKA:* Bedrucken von Kassetten und Videohüllen [s/w] ● *Kalender:* Diverse Kalenderberechnungen. U. A. Ausgabe eines Jahreskalenders auf Drucker oder Bildschirm, mit Anzeige aller Feiertage (sehr wichtig)

42. Diskette
● *FRAM 77:* Statische Berechnungen für Konstrukteure. Sehr umfangreich ● *Meßwert:* Grafische Darstellung von Meßwerten [s/w], incl. Sourcecode in Pascal

43. Diskette
● *System:* Accessoir ähnlich dem Kontrollfeld. Besonderheiten: Uhr und 4 Weckzeiten, Druckereinstellung (Rand, Schriftart,...) [s/w], incl. Sourcecode in Pascal ● *Snaphelp:* Hiermit kann SNAPSHOT (1st Word Utility) auch auf DEGAS angewendet werden, incl. Sourcecode in Pascal

44. Diskette
Spieledisk
● *TTT:* Dreidimensionales Tic Tac Toe (4★4★4) [s/w] ● *El Bozo:* El Bozos Window Puzzle ● *Senso:* Fordern Sie Ihr Gedächtnis [s/w] ● *Kalah* ● *Tri Hex:* Räumliches Tic Tac Toe (3★3★3) [s/w] ● *Börse:* Spekulieren am Börsenmarkt [s/w] ● *Solitär:* [s/w]

45. Diskette
● *Complex:* Bibliothek für komplexe Berechnung ● *Master:* Generationsberechnung mit frei definierbaren Regeln [s/w] ● *Pegasus:* Malprogramm [s/w] ● *Con_ST:* Terminalprogramm u. a.

46. Diskette
● *Termin:* Terminplanung als Accessoir [s/w] ● *KFZ-Stat:* Statistikprogramm für's Auto [s/w] ● *Lager:* Lagerverwaltung [s/w] ● *ST_Phone:* Telefonkostenberechnung [s/w]

47. Diskette
Utilitydisk
● *M_Copy_87:* Backupprogramm mit vielen Optionen und Musik ● *Copy_FMT:* Backupprogramm als Accessoir ● *Uhrzeit:* diverse Utilities zur eingebauten Uhr ● *1st_step:* Kalender und Biorhythmus [s/w] ● *Lotto:* [s/w] ● *P2_Init* u.a.

48. Diskette
● *Word 1.4:* Programm zum Erstellen eines luxuriösen Lexikons für 1st Wordplus

49. 50. 51. Diskette
● *Volksforth 83:* Die neueste Version des Forthinterpreters. Die Software zum Forthkurs. Alle drei Disketten gehören zum Forth-System.

52. Diskette
● *Molekül:* Darstellung chemischer Moleküle. In s/w und Farbe, wobei in Farbe mit 3D-Effekt. Atom, Ionen oder Van der Waalsradius. Ein Molekül kann gedreht und sogar animiert werden.

53. Diskette
● *Matrix:* Komfortables Matritzenberechnungsprogramm mit voller GEM-Steuerung (Icons, Windows, Menüzeile). Maximal 16★16 Matritzen, doppelte Genauigkeit (64 Bit). ● *LIB:* Sammlung von ST-Pascal-Bibliotheken. Incl. Source und Anleitung. Stringumwandlung (str <--> int, long_int, real) Datums+Zeitrechnungen Textformatierungen (links, rechtsbündig, zentriert, u.v.m.)

54. Diskette
Spielesammlung:
● *Bounce* Bouncing Bubbles. Schnelles Actionspiel für Joystick und Tastatur. (s/w) ● *Domino:* Gefräßige Würmer auf Nahrungssuche. (s/w) ● *Trio:* Verbessertes Tic-Tac-Toe mit Verschieben. (s/w) ● *Disc:* Die Serie der Spielautomaten reißt nicht ab. (s/w)

55. Diskette
Editorendisk
● *TOSEDIT:* Editor unter TOS. Gleichzeitiges Bearbeiten von zwei unabhängigen Texten. Reichhaltige Befehle. ● *EDIMAX:* Editor unter GEM. Mit Funktionstastenbelegung. Weiterhin zeichnen sich EDIMAX durch sehr schnelles Scrolling aus.

56. Diskette
● *Ausland:* Fremdsprechentrainer Englisch mit Sprachschatz für Fortgeschrittene. ● *Fugger:* Verwalten Sie ein Fürstentum und versorgen Sie ihr Volk. (s/w) ● *Yahtzee:* Altbekanntes Zockerspiel auf den ST umgesetzt.

57. Diskette
Wissenschaftliche Anwendungen
● *Kepler:* Grafische Darstellung von Himmelskörper-Bewegungen nach den Keplerschen Gesetzen. Variation von Masse, Radius, Zeit, potentieller und kinetischer Energie. ● *Kerne:* Der Krieg der Kerne, eine futiristische Schlacht; im Innern eines Computers suchen Rechenprogramme sich gegenseitig zu zerstören. ● *Meßwert:* Verarbeitung von Meßergebnissen und grafische Darstellung. (s/w).

58. Diskette
● *SM-Konverter:* Konvertiert SM-Manager-Texte in 1st Word (Wordplus) -Texte und natürlich auch zurück. ● *Compress:* Kompressiert beliebige Files. ● *Label:* Ändern des Diskettennamens. ● *Eti-Master:* Etikettendruck zu DB-Master ● *Urlaub:* Hilft den Urlaub zeitoptimiert zu berechnen.

59. Diskette
● *Themadat:* Assoziative Datenbank mit bis zu 200 Datensätzen.

Sonderdisks

Die folgenden Programme sind nicht Public-Domain. Sie können aber bei uns bezogen werden.

A.) – *TOS:* Die letzte Disketten-Version des TOS vom 6.2.1986. Zum fehlerfreien Betrieb des Rechners benötigen Sie diese Version, da z. B. mit älteren Versionen manche Ordner nicht gelesen werden (0 Bytes free).
Unkostenbeitrag 15,– DM incl. Diskette.

B.) – *RCS:* Das Resource Construction Set aus dem ATARI-Entwicklungspaket. Unverzichtbar bei der Erzeugung von Objektbäumen, Dialogboxen, Drop-Down Menüs. Kurzum: Ein Muß für GEM-Programmierung.
Unkostenbeitrag 15,– DM incl. Diskette.

Bis auf die Höhe des Unkostenbeitrags gelten die gleichen Versandbedingungen wie bei der Public-Domain-Software.

Versandbedingungen:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende Punkte:

- Legen Sie pro bestellter Diskette DM 10,- als Scheck bei (bei Sonderdisk DM 15,-)
- Fügen Sie dem Betrag folgende Versandkosten (Porto+Verpackung) bei: Inland DM 5,– Ausland DM 10,–
- Legen Sie unbedingt einen Aufkleber mit Ihrer vollständigen Adresse bei.
- Für ausländische Besteller sei erwähnt, daß Sie mit Euroschecks auch in DM bezahlen können.
- Bitte legen Sie keine zusätzlichen Bestellungen bei (Abo, Zeitschriften, Platinen, etc.), da dies den Vorgang erheblich verlangsamt.
- Außerdem vermerken Sie bitte auf der uns zugesandten Bestellung „Public Domain ST".
- Zu dem Kostenbeitrag müssen wir leider pro Scheck eine Scheckgebühr von DM 0,50 berechnen. Verwenden Sie deshalb nur einen Scheck, auf dem die Gesamtsumme steht (z. B. für 2 Disketten = DM 25,50)
für 4 Disketten = DM 45,50)

Auf Wunsch können Sie auch nach den alten Versandbedingungen bestellen, die wir zusätzlich aufrechterhalten.

Die Bezugsadresse lautet:

ST-Computer-Redaktion · Postfach 59 69 · 6236 Eschborn 1

Ausgewählte Kapitel der Basic-Programmierung

Die größte Anzahl der Computer-Einsteiger bevorzugen die Sprache Basic, um ihre ersten Schritte in Sachen Programmierung zu unternehmen. In dieser Serie wollen wir deshalb auf die Eigenschaften dieser Sprache eingehen, und zwar nicht als Kurs von A...Z, sondern jedesmal ein anderes Kapitel der Basic-Programmierung aufschlagen und dabei sowohl auf die Einsteiger als auch auf diejenigen eingehen, die der Sprache Basic schon mächtig sind. Daß dabei die einen oder anderen zu kurz kommen können, ist leider unvermeidlich, aber wir tun unser Bestes. In diesem ersten Teil werden wir uns ausführlich mit der Bildschirmausgabe beschäftigen, die auf den zweiten Blick schon einiges zu bieten hat. Viele Beispiele in GFA- und Omikron Basic sollen Ihnen dabei das Verständnis erleichtern.

Heraus mit den Daten?

Im Inneren eines Rechners geht einiges vor, Bits und Bytes fliegen nur so im Rechnerspeicher umher. Der Prozessor ist am Keuchen und kommt bei all den Rechnungen leicht ins Schwitzen. Die Speicherchips (RAM) werden mit für sie selbst unverständlichen Zahlen beschrieben, und nach all der Arbeit auch noch dazu aufgefordert, diese Zahlen jederzeit wieder preiszugeben. Der Festwertspeicher (ROM) ist wie ein Telefonbuch, er ist nur zum Lesen geschaffen und auf ihn wird ständig zugegriffen. Aber was interessiert das ganze Innenleben der grauen Kiste, auf das äußerlich Sichtbare kommt es an, und dazu wird meist der Bildschirm mißbraucht. Doch wie kommen die Informationen darauf?

Wenn man die Sache von der Hardware angeht, muß man sagen, daß die Hersteller eigens dafür einen weiteren Krabbelkäfer (CHIP) entwickelt haben – den Videochip. Er macht alles sichtbar, was er soll. Aber wie sagt man ihm, was er zeigen soll?

Um nun etwas auf dem Bildschirm erscheinen zu lassen, gibt es mehrere Befehle. Der erste ist PRINT. Er ist wohl allseits bekannt und auch recht einfach. Er bringt sowohl Text als auch Zahlen auf den Schirm – und dies recht einfach, wie Sie gleich sehen werden.

Programm	Ausgabe
`PRINT "Hallo mein Meister"`	*Hallo mein Meister*
`PRINT 3+4`	*7*
`A$= "Hier bin ich wieder"` `PRINT A$`	*Hier bin ich wieder*
`A=462657` `PRINT A`	*462657*
`PRINT "A"`	*A*
`PRINT SIN(1)`	*.84147098*
`PRINT "Drei plus Vier = ";3+4`	*Drei plus Vier = 7*

Generell bringt PRINT alles auf den Bildschirm, was hinter ihm steht. Dabei unterscheidet er prinzipiell zwischen Text und Zahlen. Alles, was zwischen zwei Anführungszeichen steht, also Text, wird genau so auf den Bildschirm gezaubert. Steht hinter Print eine Variable, so erscheint nicht sie, sondern ihr Inhalt auf dem Bildschirm. Als einfaches Beispiel geben wir der Variablen 'A' den Wert 462657 und lassen ihn uns anschließend ausgeben (Für Kenner: die Zahl ist nicht irgendeine Zahl, sondern spiegelt das Ergebnis eines Tennisspiels zwischen Boris Becker und Iven Lendel wieder).

Hinter PRINT können aber auch komplizierte Rechenvorgänge erscheinen; dabei wird zunächst der Ausdruck errechnet und dann das Ergebnis gePRINTet. Einfacher geht's nicht, oder?

Die Anweisungen hinter einem PRINT-Befehl können aber noch wesentlich mehr fordern. Will man mehrere Zahlen oder Textabschnitte hintereinander, also in einer Zeile ausgeben, so steht einem dabei nichts im Wege. Als einfaches Beispiel sei hier eine einfache Addition genannt. Der Rechner fragt nach den zwei Zahlen und gibt sogleich die Summe aus.

```
INPUT a
INPUT b
PRINT "Die Summe von ";a;" und ";b;" beträgt ";a+b
```

Nach der Eingabe von

7 (Return) 14 (Return)

erscheint folgendes Ergebnis:

Die Summe von 7 und 14 beträgt 21

Neu an diesem Beispiel ist das freundliche Miteinander von Zahlen und Text auf dem Bildschirm. Daß dies sehr praktisch ist und sehr oft angewendet werden kann, kann man voraussehen. Neben Text und Zahlen erscheinen auch zwei Semikolons. Diese bewirken, daß die benachbarten Ausdrücke direkt hintereinander gedruckt werden. Man kann das Semikolon meist auch weglassen, der Interpreter fügt sie dann selbst hinzu. Hauptsache, man weiß, was sie bewirken; die Arbeit kann ruhig der Rechner erledigen.

```
100 ' Demo Print AT
110 ' OMIKRON
120 '
130  CLS
140  PRINT @(12,0);"Links"
150  PRINT @(12,73);"Rechts"
160  PRINT @(0,35);"Oben"
170  PRINT @(24,35);"Unten";
180  WAIT 5
```

Listing 1

```
' Demo Print AT
' GFA
'
Cls
Print At(1,12);"Links"
Print At(73,12);"Rechts"
Print At(35,1);"Oben"
Print At(35,25);"Unten";
Pause 500
```

Listing 1a

```
110 ' SCROLLROUTINE
120 ' OMIKRON
130 ' Erzeugt eine einfache Laufschrift variabler Länge
140 ' z.B.:  Ausschnitt = 45
150 '
160 '
170 ' Jetzt kommt der Text
180 '
190 A$="Die aktuellen NEWS  ....  BONN  -  Im Bundestag nicht's Neues"
200 A$=A$+" .... OBERAMMELSBACH  -   Bauer Heums Kuh Elmeretta gab heute"
210 A$=A$+" 17 Liter Milch in des Bauerns Milchkanne "
220 A$=A$+" .... LONDON   -   Lady Di ernannte Christian Schormann zu "
230 A$=A$+" Ihrem neuen Hofmusikanten !! "
250 CLS
270 REPEAT
280   Scroll(10,10,A$,30)' x,y,Text,Ausschnitt
290 UNTIL INKEY$ <>""
320 DEF PROC Scroll(Xpos,Ypos,Satz$,Ausschnitt)
330   LOCAL A$,I,Xpos,Ypos,Ausschnitt
340   IF Ausschnitt>79-Xpos THEN Ausschnitt=79-Xpos: ENDIF
350   A$= SPACE$(Ausschnitt)+A$
360   FOR I=1 TO LEN(A$)
370      PRINT @(Ypos,Xpos); MID$(A$,I,Ausschnitt)
380   WAIT .1
390   NEXT I
400 RETURN
```

Listing 2

```
' SCROLLROUTINE
' HS
' Nun mit GFA
'
'
' Wieder brandaktuell
'
A$="Die aktuellen NEWS  ....  BONN  -  Im Bundestag immer noch nicht's Neues"
A$=A$+" .... OBERAMMELSBACH  -   Bauer Heums Kuh Elmeretta gab heute"
A$=A$+" überhaupt keine Milch, da Bauer Heums die Melkzeit verschlief"
A$=A$+" .... LONDON   -   Lady Di's neuer Liebhaber heißt Christian Schormann "
A$=A$+"  - Prinz Charles versucht sich als Hofmusikant !! "
Cls
Repeat
  Gosub Scroll(10,10,A$,30) ! x,y,Text,Ausschnitt
Until Inkey$<>""
'
Procedure Scroll(Xpos,Ypos,Satz$,Ausschnitt)
  Local I
  If Ausschnitt>79-Xpos
    Ausschnitt=79-Xpos
  Endif
  A$=Space$(Ausschnitt)+A$
  For I=1 To Len(A$)
    Print At(Ypos,Xpos);Mid$(A$,I,Ausschnitt)
    Pause 5
  Next I
Return
```

Listing 2a

Ein anderes Zeichen ist das Komma. Mit ihm hat es etwas ganz anderes auf sich. Es bewirkt nämlich, daß bei seinem Auftreten immer zum nächsten Tabulator gesprungen wird. Ein Tabulator ist in jeder achten Spalte anzutreffen. Bei einem Komma springt der Rechner also immer zum nächsten Tabulator und füllt den Platz dazwischen mit Leerzeichen auf. Hier gilt – wie fast überall in der Rechnerwelt – das Motto: Probieren geht über Studieren. Aber nehmen Sie es nicht allzu ernst, es gehört schon eine Menge Wissen dazu, aber Probieren ist trotzdem sehr hilfreich.

Natürlich fordert das Beispiel dem Rechner keine Denkverrenkungen ab, die prinzipielle Arbeitsweise wird jedoch (hoffentlich) klar.

Zum Abschluß von PRINT noch etwas anderes:

A$="Ich bin ein String und mache Bing"
PRINT A$;chr$(7);

Die PRINT-Anweisung schreibt den angegebenen Text und bringt – noch einen 'Bing' auf den Bildschirm, oder exakter ausgedrückt: auf den Lautsprecher des Bildschirms.

Da hat sich ein neuer Befehl eingeschlichen. 'CHR$' bewirkt die Umwandlung einer Zahl in einen Buchstaben. Man versuche einmal folgende Zeile.

PRINT chr$(72),chr$(65),chr$(76,chr$(76),chr$(79)

Nach dem Eintippen erscheint ein allseits bekanntes Grußwort auf dem Bildschirm, das wir aber hier nicht verraten möchten. Man sieht daran aber, daß jedem Buchstaben eine Zahl zugeordnet ist. Diese Kodierung nennt man ASCII-Code. Darin entspricht ein 'A' beispielsweise der dezimalen Zahl 65, ein 'C' der Zahl 67. Dem oben verwendeten Wert 7 ist im ASCII-Code das Bing zugeordnet – des Rätsels Lösung. Läßt man eine Schleife von '0' bis '255' laufen, wird man außer den normalen Buchstaben auch noch einige ST-Sonderzeichen sehen, zwar nicht alle, aber doch ziemlich viele.

```
For I=0 To 255
   Print chr$(i);
Next I
```

Wie man auch die anderen zu Gesicht bekommt, erläutern wir später.

PRINT Wohin?

Beim Ausprobieren werden Sie sicher schon gemerkt haben, daß die Ausgabe mehrerer PRINT-Anweisungen immer untereinander erscheinen. Der erste Text in der obersten Zeile, der Rest darunter oder – nach Abschluß eines PRINT-Befehls – mit folgendem Semikolon dahinter. Aber eine bestimmte Position ist damit nicht zu erreichen. Abhilfe schafft ein anderer Befehl. In GFA-Basic lautet dieser PRINT AT und hat folgende Syntax.

PRINT AT(Spalte,Zeile); irgendwas

Dabei kann *irgendetwas* das Gleiche sein, wie bei PRINT, also eine Zahl oder Text.

Die Wirkung ist folgende: Der Cursor wird an der angegebenen Stelle positioniert und der Text oder die Zahlen werden an diese Stelle gedruckt. Das ist für viele Anwendungen unverzichtbar, da nur so ein geordneter Bildschirmaufbau erfolgen kann. Ein einfaches Beispielprogramm zeigt die Anwendung dieses Befehls.

In OMIKRON-Basic sieht es ein wenig anders aus: Die Syntax lautet dort

PRINT (Zeile,Spalte); irgendwas

Ansonsten sind beide Befehle identisch, außer, daß die Zählung der Zeilen, bzw. Spalten in OMIKRON-Basic bei Null in GFA-Basic jedoch bei Eins beginnt, sowie die unterschiedliche Reihenfolge der Koordinatenangabe. Warum dies so ist, frage man die Programmierer der Interpreter.

Ein etwas größeres Beispiel der PRINT AT-Nutzung sehen Sie in Listing 2 und 2a. Die Programme erzeugen eine Laufschrift, die an jeder Stelle des Bildschirms und in beliebiger Länge erscheint. Es mag sein, daß der PRINT AT-Befehl in den Programmen eher eine nebenstellige Rolle spielt, aber dafür ist es eine Routine, die schon recht brauchbar ist.

Ausgabe mit Format

So einfach die bisherigen Befehle waren, so leicht erreicht man auch ihren Leistungshorizont. Und zwar dann, wenn es um formatierte Zahlenausgabe geht.

Sie werden sicherlich festgestellt haben, daß Basic immer soviele Stellen hinter dem Komma ausgibt, wie errechnet wurden. Meist ist dies aber ziemlich überflüssig, wenn nicht sogar unschön. Hat man zum Beispiel bei der Haushaltsrechnung ausgerechnet, daß im vergangenen Monat exakt DM 16,86445231 pro Tag für Süßigkeiten verprasst wurden, so sähe die Zahl aber schöner aus, wenn man sie auf DM 16,86 runden würde. Dafür gibt es nun den Befehl PRINT USING. Er sorgt dafür, daß eine Zahl genauso formatiert erscheint, wie es gewünscht wird. PRINT USING kann allerdings noch wesentlich mehr als die Abrundung der Nachkommastellen. Da wäre auch generell die Festlegung, ob eine Zahl mit oder ohne Exponent ausgegeben wird, ob das Vorzeichen erscheint oder ob nicht. Einige Beispiele sehen Sie in den Listing 3 bzw. 3a.

```
100  ' Beispiel PRINT USING
110  ' OMIKRON
120  '
130  PRINT PI *1000
140  PRINT USING "#####.###"; PI *1000
150  PRINT USING "#####.#######^^^^"; PI *1000
160  PRINT USING "+####.#######^^^^"; PI *1000
170  PRINT USING "Es kostet ####. DM und ## Pfennige ";235.76
180  PRINT USING "Bitte überweisen Sie DM *x########.##",3663.76



 3141.5926535897932
 3141.592
     3.1415926D +3
   +3.1415926D +3
Es kostet  235. DM und 76 Pfennige
Bitte überweisen Sie DM xxxxxx3663.76
```

Listing 3

```
' Beispiel PRINT USING
' GFA
'
Print Pi*1000
Print Using "#####.###",Pi*1000
Print Using "#####.#######^^^",Pi*1000
Print Using "+####.#######^^^^",Pi*1000
Print Using "Bitte überweisen Sie DM *########.##",3663.76
Print Using "\.\","ABCDEFGHIJKLMN"


3141.5926536
 3141.593
31425.9265360E+0
+3141.6926536E+03
Bitte überweisen Sie DM *****3663.76
ABC
```

Listing 3a

```
100   '  TEXTSTIL-Demo
110   '  OMIKRON
120   '
130   DATA 1,2,4,8,16,5,13,11
140   DATA 0
150   '
160   CLS
170   '
180   REPEAT
190     READ Stil
200     TEXT STYLE =Stil
210     Ypos=Ypos+30
220     TEXT 100,Ypos,"Hallo hier ist der ST"
230   UNTIL Stil=0
240   '
```

Listing 4

Interessant ist hierbei, daß beide Basic-Versionen doch einige Unterschiede zeigen. Da wäre zum einen die wesentlich höherer Genauigkeit des OMIKRON-Basics, doch auch ansonsten zeigen sich Unterschiede. So rundet GFA-Basic die Nachkommastellen auf den angegebenen Wert, während OMIKRON-Basic die nachfolgenden Stellen abschneidet. GFA-Basic zeigt recht merkwürdige Ergebnisse, bei der Ausgabe in Exponentialdarstellung, die schlichtweg falsche Ergebnisse liefert.

```
'  TEXTSTIL-Demo
'  GFA
'
Data 1,2,4,8,16,5,13,11
Data 0
'
Cls
'
Repeat
  Read Stil
  Deftext ,Stil
  Ypos=Ypos+30
  Text 100,Ypos,"Hallo hier ist der ST"
Until Stil=0
```

Listing 4a

Hallo hier ist der ST
Hallo hier ist der ST
Hallo hier ist der ST
Hallo hier ist der ST
Hallo hier ist der ST
Hallo hier ist der ST
Hallo hier ist der ST
Hallo hier ist der ST
Hallo hier ist der ST

Bild 4: Einige Schriftproben

```
110  ' Beispiel Textforma tierung
120  ' OMIKRON
130  '
140  CLS
150  TEXT STYLE =0: TEXT HEIGHT =13
160  A$="Wie ist das Wetter?"
170  BOX 95,90,215,170
180  '
190  '
200  TEXT 100,130,A$,200,1,0
210  '
220  TEXT 100,160,A$,200,0,1
230  '
240  TEXT HEIGHT =6: TEXT STYLE =4
250  '
260  TEXT 100,190,A$,200,1,0
270  '
280  TEXT 100,220,A$,200,0,1
```

Listing 5

Auch ansonsten gibt es Differenzen, aber das soll hier ja kein Basic-Vergleichstest sein – jedem das Seine. Man erkennt aber, daß dieser Befehl sehr wirkungsvoll ist.

Noch ein Tip: Besitzer eines Druckers sollten anstatt PRINT USING doch einfach einmal LPRINT USING eingeben; der Drucker freut sich.

Wie hätten Sie's gerne?

In vielen Programmen haben Sie sicher schon unterschiedliche Schriftarten bemerkt, die entweder kursiv, fett oder sonstwie erscheinen. Der ST, oder besser gesagt das GEM hat da schon Vorsorge getroffen und sdtellt diese verschiedenen Schriftformen zur Verfügung. Doch nicht nur in Form, sondern auch in der Größe sind sie frei einstellbar. In Basic gibt es nun (glücklicherweise) Befehle, mit denen diese Schriftarten und Schriftgrößen anwählbar sind und an jeder Stelle des Bildschirms plaziert werden können. Doch alles der Reihe nach.

Zunächst sei erwähnt, daß entgegen des normalen PRINT-Befehls bei TEXT, das ist der betreffende Befehl, die Ausgabe an jede beliebige Bildschirmstelle erfolgen kann, also nicht – wie bei PRINT AT – an die vorgegebenen Zeilen gebunden ist, sondern auf den vollen Bildschirmbereich erfolgen kann. Man muß hierbei beachten, daß bei Verwendung eines Farbmonitors die maximalen Koordinaten kleiner sind, die Schrift dementsprechend größer.

Die etwas Fortgeschrittenen können ja einmal versuchen, eine Laufschrift mit dem TEXT-Befehl zu erzeugen, die nicht buchstaben- sondern pixelweise scrollt.

Die Syntax ist bei OMIKRON- und GFA-Basic etwas verschieden, aber das entnehmen Sie ja den Listings hierzu. Im Grunde ist die Wirkung gleich, da beide Interpreter auf die gleiche GEM-Funktionen zurückgreifen. Diese erlaubt es, optional den angegebenen Text auf eine bestimmte Breite zu bringen. Da sich die Textbreite normalerweise aus der Breite der einzelnen Buchstaben zusammensetzt, muß man die gewünschte Breite entweder durch Dehnung der Buchstabenzwischenräume oder des Platzes zwischen den einzelnen Worten erzeugten. Man

erreicht damit leicht einen ordentlichen Bildschirmaufbau, aber schauen Sie sich doch einfach die Demos an.

Die oben angesprochenen, verschiedenartigen Textformen lassen sich auf diesen TEXT-Befehl leicht anwenden – beginnen wir mit OMIKRON Basic. Dort bestimmen die Befehle

TEXT STYLE
TEXT HEIGHT
TEXT COLOR

das Aussehen des Textes. Der Textstil kann beim ST die Werte von 0 bis 31 annehmen. Probieren Sie sie einfach einmal durch.

```
FOR I=0 to 31
   Text Style=I : Text 100,100,
   "Hallo ich bein ein Testsatz"
   Wait 0.5
Next I
```

Sie werden sehen, daß dabei manchmal recht skurrile Kombinationen entstehen. Der Grund dafür liegt darin, daß verschiedene Grundschriftarten gemischt werden. Jeder Schriftart ist ein bestimmter Wert zugeordnet. Sie kommen folgendermaßen zustande.

1 Fettschrift
2 Hellschrift
4 Kursiv
8 Unterstrichen
16 Umrandete Schrift

Will man nun z. B. eine fette Kursivschrift erzeugen, so gibt man an:

TEXT STYLE = 1+4

Somit kann jede Schriftkombination gezielt erzeugt werden.

Nach gleichem Prinzip ist die Textgröße verstellbar. Der dazugehörige Befehl lautet:

TEXT HEIGHT = 1..26

Bei Verwendung von GFA-Basic gibt es dafür nur einen Befehl. Er beinhaltet sämtliche Einstellungen der Textform. Er lautet:

DEFTEXT Farbe, Stil, Winkel, Größe

Die Variable 'Winkel' bewirkt die Textausgabe in vier verschiedenen Richtungen, also normal, nach oben, nach unten und umgedreht. 'Winkel' nimmt dabei Werte von 0 bis 3600 an. Trotzdem sind, das liegt leider am Betriebssystem des Rechners, nur vier Richtungen möglich.

Die Variable 'Farbe' gibt zusätzlich die Farbe an, in der der Text ausgege-

```
Wie    ist    das    Wetter?
Wie ist das Wetter?
Wie    ist    das    Wetter?
Wie ist das Wetter?
```

Bild 5: Exakte Breite

```
' Beispiel Textformatierung
' GFA
'
Cls
Deftext ,0,0,13
A$="Wie ist das Wetter?"
Box 95,90,300,250
'
'
Text 100,130,-200,A$
'
Text 100,160,200,A$
'
Deftext ,4,0,6
'
Text 100,190,-200,A$
'
Text 100,220,200,A$
```

Listing 5a

```
!"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~Δ
ÇüéâäàåçêëèïîìÄÅÉæÆôöòûùÿÖÜ¢£¥ßƒ
```

Bild 6: Der gesamte ST-Zeichensatz

```
' Ausgabe des ST-Zeichensatzes
'
CLS
FOR I=1 TO 255
  TEXT 100+8*(I MOD 32),100+ INT(I/32)*20, CHR$(I)
NEXT I
```

Listing 6: Zeig her deine Zeichen

```
100 ' Pfeifenkopf
110 ' OMIKRON
120 '
130 CLS
140 TEXT HEIGHT =25: TEXT STYLE =8
150 TEXT 200,50,"Pfeifenkopfdemo"
160 TEXT STYLE =0
170 '
180 FOR G=1 TO 55
190    Pfeifenkopf(G*30-250,100,G)
200 NEXT G
210 '
220 END
230 '
240 DEF PROC Pfeifenkopf(X,Y,Groesse)
250    ' Bringt den Pfeifenkopf mit verschiedenen
260    ' Groessen, irgendwo auf den Bildschirm
270    LOCAL A!
280    IF Groesse>=13 AND Groesse<=25 THEN
290       A!=8/13
300       TEXT HEIGHT =Groesse
310       TEXT X,Y, CHR$(28)
320       TEXT X+Groesse*A!,Y, CHR$(29)
330       TEXT X,Y+2*Groesse*A!, CHR$(30)
340       TEXT X+Groesse*A!,Y+2*Groesse*A!, CHR$(31)
350    ENDIF
360  RETURN
```

Listing 7



Bild 7: Raucher unter sich

```
' Pfeifenkopf
' GFA
'
Cls
Deftext ,8,0,25
Text 200,50,"Pfeifenkopfdemo"
Deftext ,0
'
For G=1 To 55
  Gosub Pfeifenkopf(G*30-250,100,G)
Next G
'
End
'
Procedure Pfeifenkopf(X,Y,Groesse)
  ' Bringt den Pfeifenkopf mit verschiedenen
  ' Groessen, irgendwo auf den Bildschirm
  Local A
  If Groesse>=13 And Groesse<=25 Then
    A=8/13
    Deftext ,,,Groesse
    Text X,Y,Chr$(28)
    Text X+Groesse*A,Y,Chr$(29)
    Text X,Y+2*Groesse*A,Chr$(30)
    Text X+Groesse*A,Y+2*Groesse*A,Chr$(31)
  Endif
Return
```

Listing 7a

ben wird. Bei monochromem Betrieb des Rechners sind natürlich nur die Farben Schwarz oder Weiß möglich.

Im Farbbetrieb allerdings vier oder sechzehn, je nachdem ob mittlere oder niedrige Auflösung eingestellt ist.

In den folgenden Listings sind einige Beispiele angeführt, die die verschiedenen Schriftarten zu 'Schirme' bringt.

Der TEXT-Befehl hat zudem noch die Eigenschaft, sämtliche Sonderzeichen des ST zu drucken, angefangen bei den Pfeilen, über die Digitalziffern, bis hin zu Wurzel etc. Die etwas komplizierte Koordinatenangabe bei diesem Textbefehl bewirkt lediglich, daß die zeichen in mehreren Zeilen ausgegeben werden. Der darin verwendete Befehl 'MOD' bewirkt, daß die erste Zahl durch die zweite dividiert wird und davon der Rest genommen wird. 'I MOD 32' bewirkt also, daß die resultierende Zahl immer von 0 bis 31 läuft und dann wieder bei Null anfängt. Somit kommt die X-Position der Zeichen zustande. Die Y-Position wird bei jedem 32sten Schleifendurchlauf um 20 erhöht.

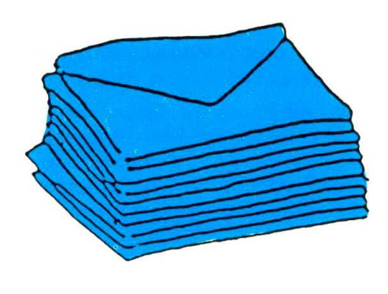
Ein Zeichen, oder besser gesagt vier, haben wir nochmal herausgegriffen und daraus ein kleines Programm gemacht. Gemeint ist der Pfeifenkopf, der sich aus den Zeichen '28,29,30,31' zusammensetzt. Das Programm kann diesen Kopf an jeder Stelle des Bildschirms zeichnen und vor allem in allen möglichen Größen. Keinesfalls sollte man den Pfeifenkopf als Sympathieerklärung mit Rauchern sehen, dafür ist ATARI verantwortlich, nicht wir.

Das Berechnen der vier Positionen, die Zeichen müssen ja schließlich zusammen passen, übernimmt die Prozedur; im eigentlichen Aufruf werden nur die Position und die Größe übergeben.

Wem die Arbeitsweise von Prozeduren zu kompliziert ist, der sollte einen Blick ins (hoffentlich) hilfreiche Handbuch werfen, das Thema Prozeduren und Funktionen wird ein andermal behandelt.

Wir hoffen, daß wir Sie ein wenig in diesen Bereich der Basic-Welt einführen konnten, und daß Ihnen die Serie einen Nutzen bringt. Für Anregungen und Vorschläge sind wir jederzeit dankbar.

(HS + MM)



Profimat-Probleme Leserecke ST 12/86, 2/87, 5/87

Sehr geehrte Damen und Herren, da das Problem mit 'move.1 #btext, –(sp)' immer noch ungelöst scheint, möchte ich hiermit auf ein paar Bus im Profimat ST hinweisen. Der Fehler liegt beim beschriebenen Problem allerdings nur zum Teil am Profimat ST. Das kleine Beispielprogramm läuft deshalb nicht, weil in der Standardeinstellung des Profimat ST, so wie er ausgeliefert wird, im Pull-Down-Menü 'Assembler' nur die Option 'PC-relativ' abgehakt ist. Das Beispielprogramm muß aber als 'relozierbar' bearbeitet werden, dann funktioniert es auch ohne Änderung. Als Assembler-Anfänger kann man das natürlich nicht wissen, und wenn man sich auf den sonst so schlauen Assembler verläßt, ist man in diesem Fall leider verlassen, denn er bemerkt diesen Fehler nicht – im Normalfall meldet er einen derartigen Fehler während des Assemblierens. Also wird das Programm als PC-relativ gespeichert, beim Laden nicht reloziert und der Absturz sichergestellt. Der Befehl PEA hingegen ist ohne Einschränkung PC-relativ erlaubt, darum läuft das Programm dann auch. Nach einigen schlaflosen Nächten mit dem Profimat ST – ein auf dem MCC Assembler funktionsfähig entwickeltes Programm lief ohne Fehlermeldung durch den Assembler, jedoch nicht auf dem Rechner – habe ich noch zwei weitere Fehler gefunden:

1. Aus 'move.w #44,$ffff8802' macht der Assembler 'move.w #44,$8802', was der Debugger auch noch mit 'move.w #44, –$77fe' übersetzt. Man hätte wohl beim Zieloperanden auf die Übersetzung des höchstwertigen Bits als '–' verzichten sollen, da dort meist nur Register oder Adressen stehen. (Ausnahme: ...,–8(sp) und ähnliche)
   Abhilfe: move.w #44,$ff8802, der Adressbus ist eh' nicht breiter.
2. Aus 'eor.b #7,d5' (zugegebenermaßen nicht sauber ausgedrückt, aber der MCC versteht es, und der Profimat ST will ja besonders narrensicher sein) macht der Assembler 'eor.b d4,d5'. Zusätzlich kommt er mit seinen Opcode-Längen durcheinander, der letztgenannte Befehl ist zwei Bytes kürzer als der Richtige, damit sind zwei Müllbytes (00 07) bis zum nächsten Befehl – beim Laden wird es der Relocater danken, vom PC während des Programmlaufs ganz zu schweigen...
   Abhilf: eori.b #7,d5.

Christian Suder, 1000 Berlin 47

CP80 und CP80-x am ST

Lieber Gernot H. in Linz (und andere, die aus der C64-Zeit den CP80-x-Drucker von Melchers besitzen), ich betreibe meinen Drucker mit einem zusätzlichen selbstgestrickten Anschlußkabel wechselnd am C64 und am Atari ST. Ermöglicht wird das durch den Einbau eines Schalters in die Drucker-Rückwand. Dorthin werden drei Drähte von dem für die Umschaltung zwischen seriell- und parallel-Modus zuständigen DIL-Schalter herausgeführt, weil dieser leider sehr versteckt angeordnet ist.
Durch Betätigen des Schalters sowie Umstecken des Anschlußkabels kann jeweils der eine oder andere Computer an den Drucker angeschlossen werden.
Auch die Hardcopy-Funktion habe ich aktiviert, im Oktoberheft '86 steht dazu einiges, allerdings für den CP80 (ohne X) Zutreffendes. Die Pokes für das TOS von Diskette lauten für den CP80-x:

```
Poke $9D56 , 0000
Poke $9DB6 , 0000
Poke $17AC2, 6Cff
```

Mit dem Disk-Doctor kann man die Bytefolgen im File „TOS.IMG" suchen lassen und ändern. Bei mir klappt das.

```
Alt:
8A 3E BC 00 0A 61
2A 3E BC 00 0A 61
1B 59 FF 00 1B 33
```

```
Neu:
8A 3E BC 00 00 61   Linefeed
2A 3E BC 00 00 61   Linefeed
1B 6C FF 00 1B 33   Bitmapmodus
```

Ralf Störmer, 6730 Speyer

Frage zu benutzerdefinierten Objekten auf dem Atari ST unter Modula-2

Sehr geehrte Damen und Herren!
Der Atari ST ermöglicht die Verwendung von Benutzer definierten Objekten, mit deren Hilfe man das Manko der nicht vorhandenen runden Objekte durch eigene Routinen ausgleichen könnte.
Ich arbeite mit dem Modula 2 Compiler von TDI, der die dazu notwendigen Strukturen (ApplBlk, Param Blk) auch zur Verfügung stellt. Die Startadresse der zum benutzerdefinierten Objekt gehörenden Prozedur erhalte ich auch (wie ich meine) ohne Probleme durch den Cast-Operator ADDRESS(Zeichenprozedur). Sobald die ObjectDraw jedoch auf den benutzerdefinierten Typ stößt, bricht das Programm mit einer Fehlermeldung (Address Error Trap at $FDD xxx) ab.
Die Richtigkeit der Zeiger habe ich mehrfach, zuletzt mit dem SID im Einzelschrittmodus überprüft. Letzterer zeigt dieselbe Fehlermeldung nach der Ausführung der Trap#2 Anweisung.
Liegen Ihnen da irgendwelche Erfahrungen vor? Ich wäre Ihnen dankbar für eine baldige Antwort. Sollte Ihnen ein Weg bekannt sein, bzw. Sie auf meine Anregung hin einen finden, bitte ich mir eine Kopie des jeweiligen Listings zukommen zu lassen. Notfalls auch auf Diskette, die eventuell anfallenden Kosten übernehme ich selbstverständlich.
Ähnliche Probleme haben vielleicht auch Benutzer des ST Pascal Plus, deren Handbuch diesen Objekttyp allerdings ignoriert?
F. Neudecker, 62 Wiesbaden-Bierstadt

GFA-BASIC

Sehr geehrte Damen und Herren, ich programmiere derzeit sehr intensiv in GFA-BASIC und habe gerade zwei Programmprojekte parallel laufen. Die Länge der Programme bewegt sich in nicht kompiliertem Zustand zwischen 80 und 90 Kiloby-

tes. Nun ist es mir jetzt schon bei beiden Programmen vorgekommen, daß sie immer kurz nach dem Start entweder abstürzten oder der Interpreter ohne Unterlaß 2-, 3- oder 4-(!) Bomben-Errors meldete (– > Resetknopf).

Da es ähnliche Probleme damals auch schon auf meinem 800 XL in BASIC gab, versuchte ich, eine alte Methode, die man in solchen Fällen auf diesem 8-Bit-Computer z. T. anwendet, auch auf meinem ST auszuprobieren:

1) Interpreter laden
2) Programm ganz normal mit LOAD einladen
3) Programm gleich wieder mit SAVE,A (auf RAM-Disk) abspeichern
4) Interpreter verlassen
5) Interpreter wieder laden
6) Mit SAVE,A abgespeichertes Programm mit MERGE von RAM-Disk einladen (Geduld – es kann ziemlich lange dauern!)
7) Programm nun mit SAVE wieder auf Diskette abspeichern
8) Interpreter verlassen

Und siehe da: Das ca. 85 KB lange Programm ist um ca. 1500 – 1600 Bytes in seiner Länge geschrumpft!

Nun läuft das Programm auch wieder.

Vermutlich lag das Nichtfunktionieren des Programms daran, daß der Interpreter einmal z. B. im Direktmodus zu Testzwecken benutzte Variablen nicht mehr aus seinem Variablenverzeichnis löscht. Zumindest steht der Variablenpointer einer Variablen nach dem Clear- bzw. Clr- bzw. Erase-Befehl noch auf dem alten Wert.

Danach wird wahrscheinlich jeder Variableneintrag – auch, wenn es diee Variable im Programm eigentlich gar nicht gibt – bei jedem SAVE-Befehl mit abgespeichert, was nach einiger Zeit zu dem oben beschriebenen Kollaps führen kann, da auch alle schon sinnlos gewordenen Einträge des Programms mit dem LOAD-Befehl wieder aufgenommen werden und sich die Einträge mit der Zeit immer weiter anhäufen.

Bei dem SAVE,A/MERGE-"Heil"-Verfahren wird jedoch eine neue Variablentabelle von nur den Variablen erstellt, die auch tatsächlich im Programm vorkommen.

Rainer Haseneder, 7100 Heilbronn

1st Word Plus

Sehr geehrte Damen und Herren!
Kritisch äußern möchten wir uns zu Ihrem Softwaretest in der ST-Computer 4/April 1987. Leider blieben in Ihrem Test einige Macken des 1st Word Plus (Version 1.89) verborgen. Wir arbeiten seit einem Monat kommerziell mit diesem Programm, gehen also nicht gerade sanft damit um. Hier nun unser „Testergebnis“:

Das die deutsche Rechtschreibung gerade durch Leistungskurse der gymnasialen Oberstufe problematisch ist, ist bekannt. Daß aber im Wörterbuch gravierende Fehler sind, ist alles andere als eine Reklame für den Autor des selbigen. „Selbstständig“ schreibt sich auch im Zeitalter der ST-Serie nur mit einem st, um nur ein Beispiel aus dem Wörterbuch zu nennen. Eine weitere Anschauung der Rechtschreibkunst ist der Menüpunkt „Wörterbuch bearbeiten“.

Problematisch wird das Laden von ASCII-Files bzw. von mit 1st Word erstellten Texten. Behauptet ein Textfenster bei manchen Files, es handele sich nicht um ASCII- bzw. GST-Format. Will man das Hinweisfenster schließen, erscheinen je nach Laune die netten Bömbchen. Einlesen der Files ist mit dem alten 1st Word ohne Probleme möglich.

Wechselt man während der Arbeit die Diskette, um einen weiteren Text von einer anderen Disk zu laden, wird der Inhalt von Ordnern nicht angezeigt, so daß ein Ausstieg aus 1st Word Plus nötig ist bzw. die benötigten Textfiles dürfen sich von Beginn an nicht in Ordnern befinden.

Wie man 1st Word Plus zum abstürzen bringt? Nichts leichter als das. Versucht man ein Wort beginnend mit dem Umlaut Ä bzw. ä ins Wörterbuch einzutragen, ist alle bis dahin investierte Arbeit vernichtet. Die anderen Umlaute am Wortanfang bereiten dem deutschen Wörterbuch ebenso Probleme, jedoch bleiben hier die netten Bömbchen aus.

Bei hastiger Befehlsauswahl bzw. mehrfacher falscher Auswahl in der Menüleiste kann es passieren, daß sich das Programm „aufhängt“. Bei nachfolgender Tastatureingabe passiert zunächst nichts, erst bei Bewegung der Maus erscheinen seltsame Zeichenfolgen. Glücklicherweise kann manchmal (leider nicht immer) noch ein Befehl (z. B. Speichern) angewählt werden, bevor nichts mehr geht. Hat man vergessen, bei „Öffnen“ eine Diskette in das Laufwerk zu legen, stürzt der Rechner nach Betätigen von „ABBRUCH“ im entsprechenden Hinweisfenster ab.

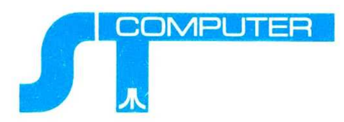
Eine Variierung des Zeilenabstandes innerhalb des Textes ist leider immer noch nur mit selbst definierten Steuerzeichen möglich, sollte aber bei einem vernünftigen Textverarbeitungsprogramm ebenso wie Proportionalschrift Standard sein. Die manuelle Voreinstellung des Trenngrades bzw. die Möglichkeit, Trennzeichen direkt in den Text eingeben zu können, die beim Umformatieren mit berücksichtigt werden, gehört auch zu einer Textverarbeitung, die professionellen Ansprüchen gerecht werden will. Auch ist die Rechtschreibüberprüfung umständlich (insbesondere das Ergänzen des Wörterbuches). Desweiteren wäre die Möglichkeit der Spaltenbildung eine weitere wichtige Funktion, die 1st Word Plus bisher leider nicht bietet.

Die Diskettenoperationen wie „Speichern“ oder „Öffnen“ sind nicht von dem dem Atari ST gebührenden Tempo. Wünschenswert wäre zudem, daß „Speichern & weiter“ im Hintergrund ablaufen würde. Gerade durch regelmäßiges Sichern des gerade zu bearbeitenden Textes (in unserem Fall zusammenhängend bis über 150 DIN A4 Seiten) geht (teure) Zeit verloren.

Abschließend ist zu sagen, daß 1st Word Plus nicht zuletzt wegen der einfachen Bedienbarkeit und des geringen Anschaffungspreises trotz der zuvor beschriebenen Mängel und Erweiterungswünsche recht brauchbar und zu empfehlen ist. Trotzdem wäre es wünschenswert, auch diese würden in einer überarbeiteten Version noch behoben bzw. fehlende Funktionen würden noch ergänzt bzw. erweitert, damit aus dem Atari ST in jeder Hinsicht ein echter „Profi-Computer“ wird, der mit Hilfe des sehr guten Monitors, seines umfangreichen Speichers, einer demnächst verbesserten Tastatur (Anschlag), und der Verarbeitungsgeschwindigkeit es auch mit viel teureren Textverarbeitungssystemen im kommerziellen Bereich aufnehmen kann.

Elisabeth Penke, 4790 Paderborn

In der nächsten



lesen Sie unter anderem

Der Sport der Denker

Schachprogramme im Überblick

Schachprogramme gibt es auf dem ST schon recht viele. Aber wie leistungsstark sind sie? Halten sie den Vergleich mit anderen System stand? Sind sie ernstzunehmende Gegner für Schachspieler? Interessante Antworten im nächsten Heft.

Setzerei in den eigenen vier Wänden

ST-TEX – mehr als ein Textprogramm

Wer für die Diplom- oder Doktorarbeiten oder sogar für ein eigenes Buch ein Textsystem benötigt, das mehr kann, als die in diesem Test beschriebenen, der sollte sich mit ST-TEX anfreunden. Mehr darüber im nächsten Heft.

Im Musiknetz geht's hoch her

Neues aus der Musikszene

Ist der ST eventuell kein Computer, sondern eine Musikmaschine? Bei der Vielzahl von guten MIDI-Programmen, könnte man dies fast glauben. Ein Überblick über Neuerscheinungen aus unserer Musikecke. Außerdem lesen Sie einen ausführlichen Artikel über die Grundlagen der MIDI-Technik.

Inserentenverzeichnis

Impressum

ST-Computer

Herausgeber:
Heim Fachverlag,
Heidelberger Landstraße 194,
6100 Darmstadt 13,
Telefon (0 61 51) 5 60 57 FAX 06151 / 55689

Verlagsleitung:
Hans-Jörg Heim

Redaktion:
Uwe Bärtels (UB Chefredakteur)
Marcelo Merino (MM)
Harald Egel (HE)
Dipl.-Ing. Harald Schneider (HS)

Adresse:
ST-Computer Redaktion
Schwalbacher Str. 64
6236 Eschborn
Tel. 0 61 96 / 48 21 58 FAX 06196 / 41137
oder
Postfach 59 69, 6236 Eschborn

Redaktionelle Mitarbeiter:
Markus Nerding (MN), Oliver Joppich (OJ)
Uli Eickmann (UE), Stefan Höhn (SH)
Jürgen Leonhard (JL), Chr. Schormann (CS)
Jörg Wilhelm (JW)

Autoren dieser Ausgabe:
C. Lippert
P. Monadjemi, C. Hahn
D. Rabich, Dr. K. Sarnow
M. Spehr, Dr. V. Kurz
U. Oehler, U. Stumm

Produktion:
Klaus Schultheis (Ltg.),
Bernd Uwe Failer
Susanne Failer
Bela Kumar
Carola Schwarze
Martina Simacek

Anzeigenverkaufsleitung:
Uwe Heim

Anzeigenpreise:
nach Preisliste Nr. 2, gültig ab 1.7.86

Vertrieb:
Hans-Jörg Heim, Uwe Heim,
Heide Schultheis

Erscheinungsweise:
11 x jährlich

Bezugspreis:
Einzelheft DM 6,–.
Jahresabonnement DM 60,– inklusive der gesetzlichen Mehrwertsteuer und den Zustellgebühren für 11 Ausgaben.
Ausland: 80,– DM inkl. Versand

Bezugsmöglichkeiten:
ATARI-Fachhändler, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag unter obiger Adresse.

Druck:
Ferling Druck Darmstadt

Manuskripteinsendungen:
Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit ihrer Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern im Heim Verlag. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

Titelseite: Fabian & Mayer
Foto: Rainer Spirandelli

hardware
software
organisation
service

CSF GMBH

Heeper Str. 106–108, 4800 Bielefeld 1, 05 21/6 16 63

Kein Kabelsalat mehr mit dem Gehäuse für ATARI ST

- **Zentrale Stromversorgung für alle Geräte einschl. 2 Drucker**
- **Einbaumöglichkeit von 2 Diskettenlaufwerken**
- **Rechner (Tastatur) kann komplett unter das Gehäuse geschoben werden (Staubschutz)**
- **Massives Blechgehäuse**

ATARI ST-Gehäuse erhalten Sie bei den autorisierten Fachhändlern



DM 149,–

2 Megabyte Speichererweiterung für ATARI ist demnächst lieferbar!

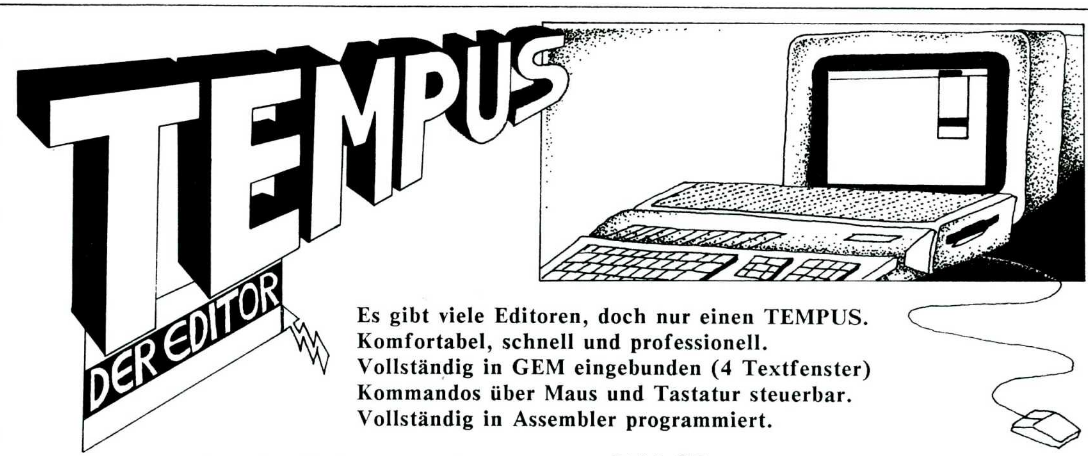
NEUES VON CSF – Speichererweiterung ohne Probleme!

- **steckbar** (ohne jegliche **Lötarbeiten**) – läuft auch auf dem 520 STM
- enorme Zeitersparnis durch einfache, bebilderte Einbauanleitung
- kein Flimmern nach der Erweiterung (durch separate, geglättete Spannung an der zweiten RAM-Bank)
- sensationeller Preis
- Bei Bestellungen bitte angeben: ☐ Speichererweiterung steckbar ☐ Speichererweiterung zum Löten

NEU: DM 198,–

Zu beziehen: **Direkt bei CSF, Bielefeld Tel. 05 21/6 16 63** **Bei allen ATARI-Händlern**	In der Schweiz: **SENN Computer AG Langstr. 31 · CH-8021 Zürich Tel. 01-241 73 73**	In Österreich: **Institut für Datenverarbeitung und Organ. Ges.mbH Rehberger Hauptstr. 95 · A-3503 Krems Tel. 0 27 32-7 05 81**

Alle Preise sind unverb. empf. Verkaufspreise



Es gibt viele Editoren, doch nur einen TEMPUS.
Komfortabel, schnell und professionell.
Vollständig in GEM eingebunden (4 Textfenster)
Kommandos über Maus und Tastatur steuerbar.
Vollständig in Assembler programmiert.

Preis bei Vorkasse im Inland: = **DM 85,--** (inkl. DM 6,-- Versandkosten)
Preis per Nachnahme im Inland: = **DM 90,--** (inkl. DM 11,-- Versandkosten)
Preis bei Vorkasse europ.Ausland: = **DM 90,--** (inkl. DM 11,-- Versandkosten)
Preis bei Nachnahme europ.Ausland: = **DM 95,--** (inkl. DM 16,-- Versandkosten)

Sollten Sie noch weitere Fragen haben, rufen Sie uns einfach an!

W. & H. Computerhandels GmbH
Förstergasse 6/3/2
A-1020 Wien
✆ 02 22 - 338 45 92

CCD

Creative Computer Design

Dirk Beyelstein - Burgstraße 9 - D-6228 Eltville - Tel.: 06123 / 16 38